



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

ÚSTAV VÝROBNÍCH STROJŮ, SYSTÉMŮ A ROBOTIKY

INSTITUTE OF PRODUCTION MACHINES, SYSTEMS AND ROBOTICS

**ON-LINE KOREKCE DRÁHY ROBOTU NA ZÁKLADĚ DAT
ZE SNÍMAČE VZDÁLENOSTI**

ON-LINE PATH CORRECTION OF THE ROBOT BASED ON THE SENSOR DATA

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Michal Ržonca

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Tomáš Kubela

BRNO 2021

Zadání diplomové práce

Ústav: Ústav výrobních strojů, systémů a robotiky
Student: **Bc. Michal Ržonca**
Studijní program: Strojní inženýrství
Studijní obor: Výrobní stroje, systémy a roboty
Vedoucí práce: **Ing. Tomáš Kubela**
Akademický rok: 2020/21

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma diplomové práce:

On–line korekce dráhy robotu na základě dat ze snímače vzdálenosti

Stručná charakteristika problematiky úkolu:

Cílem je navrhnout a implementovat SW pro on–line korekci dráhy robotu na základě senzorických dat poskytovaných snímačem vzdálenosti, který je součástí koncového efektoru daného robotu. V práci bude využit průmyslový robot KUKA s rozšířením RSI (Robot Sensor Interface) a snímač vzdálenosti Keyence. Návrh a implementace hlavního řídicího SW bude probíhat na SW platformě Twincat PLC systémů Beckhoff.

Cíle diplomové práce:

Systémový rozbor zadané problematiky.
Navrhnout SW pro řízení dráhy robotu na základě senzorických dat.
Navrhnout uživatelské rozhraní pro řízení dané úlohy.
Implementovat navržený SW.
Ověřit funkčnost SW na reálném robotickém pracovišti.

Seznam doporučené literatury:

NOF, Y. Shimon, ed. Springer Handbook of Automation. Springer-Verlag Berlin Heidelberg, 2009. 1812 s. ISBN 978-3-540-78830-0.

SICILIANO, Bruno, KHATIB, Oussama, ed. Springer Handbook of Robotics. Springer-Verlag New York, Inc., 2008. 1611 s. ISBN 978-3-540-23957-4.

Beckhoff [online]. Beckhoff Information System: 2019 [cit.: 16.9.2019]. Dostupné z: <http://infosys.beckhoff.com/>

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2020/21

V Brně, dne

L. S.

doc. Ing. Petr Blecha, Ph.D.
ředitel ústavu

doc. Ing. Jaroslav Katolický, Ph.D.
děkan fakulty

ABSTRAKT

Hlavným cieľom tejto diplomovej práce je aplikovať online korekcie dráhy robota na základe dát zo snímača vzdialenosti. V prvej časti oboznamuje čitateľa s dostupnými riešeniami, ktoré používajú najväčší výrobcovia komerčne dostupných manipulátorov. Následne sa zaoberá zdôvodnením výberu konkrétneho riešenia a popisu reálneho pracoviska. V druhej časti tejto práce sa zaoberá implementáciou korekcie dráhy, najskôr na základe generovaného signálu a následne aj podľa reálnych dát o vzdialenosti zo snímača keyence. Cieľom práce je aj vytvoriť HMI na externom PLC, ktoré bude zobrazovať požadované údaje o priebehu korekcie. V poslednej časti je vyhodnotenie nameraných údajov a priblíženie základných problémov tohoto riešenia

Kľúčové slová

On-line korekcie dráhy, real-time korekcie, Kuka KR16, Kuka RSI, Keyence IL-300, Keyence IL-1000, RSI Visual, RSI monitor

ABSTRACT

The main aim of this master's thesis is to provide online path correction of 6 – axis robot leaded by an external distance measuring sensor interface. First part is describing the actual state and required options of some robot manufacturers. Next part justifies selection and is describing the real robotic cell. Third part is focused on programming of path correction of Kuka KR16 based on generated signal and also based on signal from distance sensor interface. There is also described the communication between PLC and robot and how to display data from robot to HMI. Last part is focused on data analysis and is describing some problems discovered by application of path correction.

Key words

On-line path correction, real-time correction, Kuka KR16, Kuka RSI, Keyence IL-300, Keyence IL-1000, RSI Visual, RSI Monitor

BIBLIGRAFICKÁ CITÁCIA

RŽONCA, Michal. *On-line korekce dráhy robotu na základě dat ze snímače vzdálenosti*. Brno, 2021. Dostupné také z: <https://www.vutbr.cz/studenti/zav-prace/detail/134887>. Diplomová práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství, Ústav výrobních strojů, systémů a robotiky. Vedoucí práce Tomáš Kubela.

PREHLÁSENIE

Prehlasujem, že som diplomovú prácu na tému Online korekce dráhy robota na základě dat ze snímače vzdálenosti vypracoval samostatne s použitím odbornej literatúry a prameňov, uvedených v zozname, ktorý tvorí prílohu tejto práce.

V Považskej Bystrici dňa

Bc. Michal Ržonca

POĎAKOVANIE

Ďakujem týmto Ing. Tomášovi Kubelovi za cenné pripomienky a rady, ktoré mi poskytol pri vypracovaní záverečnej práce a za ochotu a trpezlivosť pri testovaní a nastavovaní pracoviska.

Bc. Michal Ržonca

Obsah

ÚVOD	10
1. Motivácia	11
2. Prehľad dostupných opcíí pre korekcie dráhy	12
2.1. Vymedzenie základných pojmov použitých v tejto časti	12
2.2. Opcia od výrobcu KUKA - KUKA RSI	13
2.2.1. Úvod	13
2.2.2. Základné termíny používané výrobcom KUKA	14
2.2.3. Vstupy a výstupy cez I/O systém	14
2.2.4. Vstupy a výstupy cez ETHERNET	15
2.2.5. Real-Time odpoveď	16
2.2.6. Možnosti korekcie	16
2.2.7. Safety - bezpečnosť	16
2.2.8. Systémové požiadavky pre využitie Kuka RSI	16
2.2.9. Inštalácia, aktualizácia a odinštalovanie RSI do manipulátoru, inštalácia RSIVisual na externé PC	17
2.2.10. Konfigurácia sieťového spojenia	17
2.2.11. Nastavenie RSI modulu	17
2.2.12. Základný prehľad KUKA RSI príkazov	17
2.3. Opcia od výrobcu Stäubli – Stäubli Alter	18
2.3.1. Úvod	18
2.3.2. Safety - Bezpečnosť	19
2.3.3. Obmedzenia	19
2.3.4. Základné robotické príkazy pre korekciu dráhy	20
2.4. Opcia od výrobcu ABB – ABB Externally Guided Motion	20
2.4.1. Úvod	20
2.4.2. Základný postup pri vytváraní korigovanej dráhy EGM	21
2.4.3. Základné stavy EGM a ich význam pre korekciu dráhy	21
2.4.4. Možnosti komunikácie s EGM	22
2.4.5. Základné príkazy EGM	22
2.4.6. Zhodnotenie vlastností EGM	23
3. Porovnanie jednotlivých opcíí a zdôvodnenie výberu zvoleného riešenia	25
4. Aplikácia korekcií na reálnom pracovisku	26
4.1. Úvod – zoznámenie s pracoviskom	26
4.1.1. Snímač Keyence IL-300	27
4.1.2. Zosilňovač IL-1000	28

4.2.	Zapojenie a sprevádzkovanie pracoviska	29
4.2.1.	Nastavenie výstupného signálu zo snímača IL-1000	31
4.3.	Vytvorenie a nahratie programu pre korekcie dráhy podľa sínusového signálu	31
4.3.1.	Program v RSI	31
4.3.2.	Robotický program	32
4.4.	Vytvorenie a nahratie programu na korekcie dráhy podľa dát zo snímača (snímač umiestnený paralelne ku smeru pohybu)	33
4.4.1.	Program v RSI	34
4.4.2.	Robotický program	36
4.5.	Doplnenie programu o HMI	38
4.6.	Vytvorenie a nahratie programu na korekcie dráhy podľa dát zo snímača (snímač umiestnený pred TCP)	42
4.6.1.	Program v RSI	43
4.6.2.	Robotický program	43
4.6.3.	Úprava programu pre dlhšie dráhy	45
5.	Vyhodnotenie údajov z RSI Monitoru	47
	Záver a zhodnotenie výsledkov práce	49
	ZOZNAM POUŽITÝCH ZDROJOV	50
	ZOZNAM POUŽITÝCH SYMBOLOV, JEDNOTIEK A SKRATIEK	51
	ZOZNAM OBRÁZKOV	52
	ZOZNAM TABULIEK	53
	ZOZNAM GRAFOV	54
	ZOZNAM PRÍLOH	55

ÚVOD

Základným stavebným prvkom nových veľkých tovární sú už dávno manipulátori, ktoré nahrádzajú ľudskú činnosť. Obrovské uplatnenie nachádzajú vo veľkokapacitnej výrobe, ako napríklad v automobilkách, kde vyrábajú tie isté kusy stále dookola. Obľúbené sú aj napríklad pri manipulácii v nebezpečných priestoroch, ako napríklad pri lisoch, v lakovniach alebo ako zvráťacie roboty, kde nie je práca úplne bezpečná pre človeka. Tak isto sú vhodné pre manipuláciu s ťažkými predmetmi alebo tam kde je vyžadovaná rýchla a presná manipulácia. Cieľom výrobcov, ale aj programátorov je dosiahnuť čo najvyššiu mieru automatizácie a autonómie manipulátorov. V praxi patria medzi najviac rozšírené 6-osé manipulátori, ktoré je možné doplniť rôznymi zariadeniami, ako napríklad zvráťacími hlaviciami, lakovacími hlaviciami, nitovacími kliešťami alebo hocikým zariadením zo širokej ponuky výrobcov. Tieto roboty majú väčšinou 6 rotačných osí (A1-A6). Obľúbené sú tiež 3- osé SCARA roboty, ktoré sú špecifické obrovskou rýchlosťou a sú vhodné najmä na manipuláciu s ľahkými predmetmi, pretože nemajú až takú veľkú najvyššiu manipulačnú hmotnosť. Tieto roboty majú 2 rotačné a jednu posuvnú osu. Veľmi často sa používajú napríklad na manipuláciu s čipmi na výrobnom páse, pri ich ukladaní do konečného balenia a podobných manipulačných procesoch. Tú sú tiež používané manipulátori s paralelnou kinematikou, veľmi často s 3 posuvnými osami. Hlavným obsahom mojej diplomovej práce bude spracovať možnosti autonómnej korekcie dráhy na základe dát zo snímača vzdialenosti. Pokúsim sa opísať možné dostupné opcie a následne sa pokúsim spracovať program pre korekcie dráhy. Takýto autonómny prístup môže byť vhodnou alternatívou napríklad pre nanášanie rôznych nástrekov, povlakov, alebo pri manipulácii na páse pohybujúcom sa rôznymi rýchlosťami. Rovnako bude cieľom mojej práce komunikácia a výmena dát medzi kontrolérom manipulátora a PLC, ktoré si majú medzi sebou zdieľať rôzne údaje a tieto údaje sa majú zobrazovať na obrazovke PLC. S týmto je tak isto možné stretnúť sa veľmi často v praxi, kedy práve externé PLC veľmi výrazne rozširuje možnosti manipulátora, či už z hľadiska riadenia pohybov, alebo a kontroly a predávania údajov do vyššej úrovne, alebo práce s rôznymi databázami. Rovnako je možné, aby PLC robilo úpravy dráh, alebo ich časovo oneskorilo v prípade hrozacej kolízie. PLC môže teda vo výrobe plniť aj dozornú funkciu.

1. Motivácia

Jednou z hlavných motivácií pre aplikácie online korekcií dráhy je rozhodne vyššia miera adaptability výrobného procesu a samozrejme vyššia miera autonómnosti. Ďalšou výhodou je jednoznačne možnosť predísť kolíziám. Pri pracoviskách, u ktorých je možné korigovať dráhu real time je možné nie len oddialiť nástroj v prípade, že by došlo k jeho extrémnemu priblíženiu v mieste kde to nie je žiadúce, ktoré by mohlo byť spôsobené neočakávaným pohybom obrobku, ale aj do istej miery v pokračovaní v činnosti s upravenými parametrami. Toto môže značne ušetriť náklady firmy čiastočným odstránením poškodených kusov, ale aj znížením výpadkov výroby a prispieť tak k lepšiemu chodu fabriky. Jednoznačne to môže mať vplyv aj na väčšie nasadzovanie robotov v praxi, napríklad v prípadoch kde nie sú diely úplne presné, prípadne môže dôjsť k obmedzeniu pohybu (pri manipulácii za pochodu. Rovnako to môže zvýšiť flexibilitu výrobkov, pokiaľ by boli napríklad na ich povrch nanášané nástreky, môže sa robot približovať a oddaľovať podľa potreby bez nutného prednastavenia. Mojou motiváciou k výberu tejto témy bola jej pútavosť a do istej miery získavanie skúseností z oblasti, ktorú sme nemali možnosť preskúmať počas výuky.

2. Prehľad dostupných opcií pre korekcie dráhy

2.1. Vymedzenie základných pojmov použitých v tejto časti

Ethernet

Technológia vyvinutá pre LAN (Local Area Network) počítačové siete. Jej vývoj sa začal v Palo Alto Resarch Technology v tíme pod vedením Roberta Metcalfa. Slúži na komunikáciu počítačov v sieti a každý počítač má globálne jedinečnú 48-bitovú MAC adresu. Tá je už pri výrobe implementovaná do sieťovej karty a tým je zaručená jej jedinečnosť v sieti. Pre veľkú obľubu sú tieto karty často implementované priamo do matičných dosiek. Od roku 1980 je pod spoločnosťou IEEE štandardizovaná pod štandardom IEEE 802.3 maximálna prenosová rýchlosť je do 100 Gb/s a je možné používať ako zbernicovú, tak aj hviezdicovú topológiu. Podľa použitého kábla môže byť pripojený konektorom štandardu 10base5, 10base2 alebo krútenou dvojlinkou 10baseT, častejšie označovaný ako RJ-45.

Zbernicová topológia – anglicky Bus Topology je také zapojenie siete, kedy sú všetky zariadenia v sieti pripojené na jedno prenosové médium, zbernicu. Výhodami je ľahké realizácia a jednoduché pripojenie nových zariadení, nevýhodou zase zložité hľadanie a odstraňovanie chýb

Hviezdicová topológia – anglicky Star Topology predstavuje jeden z najrozšírenejších spôsobov zapojenia dnešných sietí. Skladá sa z jedného hlavného uzlu, ku ktorému sú pripojené jednotlivé zariadenia. Všetka komunikácia ide cez tento hlavný uzol. Výhodou je jednoduchá kontrola komunikácie, možnosti delenia úrovni a prístupov. Nevýhodou je že rýchlosť komunikácie zariadení v sieti je závislá najmä na rýchlosti hlavného uzlu.

Digitálny signál

Jedná sa o nespojitý signál v hodnote aj v čase, najčastejšie je vyjadrený binárnym kódom a jednotlivé stavy sú vyjadrené 1- prúd/napätie tečie a 0 – prúd/napätie netečie. Rozmach digitálnych systémov prišiel spolu s nástupom počítačovej techniky a v dnešnej dobe sú aj niektoré analógové signály prevádzané na digitálna, táto skutočnosť je označovaná ako digitalizácia. Tá je však často spájaná s čiastočnou stratou kvality, napríklad pri prenose a uschovávaní zvuku a z toho dôvodu sú stále využívané aj analógové systémy a nosiče.

Analógový signál

Analógový signál je daný spojitými funkciami v čase, alebo čiastočne spojitými funkciami v čase. Príkladom takéhoto signálu sú napríklad akustické signály, optické signály, elektrické signály.

XML – Extensible Markup Language – predstavuje štandard pre vytváranie dokumentov, ktoré majú špecifickú stromovú štruktúru a je špecifický tým, že okrem dobrej strojovej čitateľnosti je aj prívetivý pre človeka.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<RSIOjects xsi:noNamespaceSchemaLocation="/Roboter/Config/System/Common/Schemes/RSIContext.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <RSIOject ObjType="ANIN" ObjTypeID="30" ObjID="ANIN1">
    <Parameters>
      <Parameter Name="Index" ParamID="1" ParamValue="7" />
    </Parameters>
  </RSIOject>
  <RSIOject ObjType="SUM" ObjTypeID="31" ObjID="SUM1">
    <Inputs>
      <Input InIdx="1" OutObjID="P1" OutIdx="1" />
    </Inputs>
    <Parameters>
      <Parameter Name="cVal" ParamID="1" ParamValue="-140" />
    </Parameters>
  </RSIOject>
</RSIOjects>
```

Obrázok 1 - ukážka .xml kódu RSI pre online korekcie

UDP – User Datagram Protokol – predstavuje nezabezpečený protokol na prenos datagramov medzi zariadeniami v sieti. Na rozdiel od TCP neobsahuje spätnú kontrolu, napríklad či boli všetky časti doručené alebo či neboli niektoré doručené viac krát. Kvôli tomu je vhodný najmä na časovo rýchlu komunikáciu (real-time) po zabezpečenej a malej sieti. Kvôli svojej jednoduchosti nie je tak náročný na spracovanie.

IP – Internet Protokol – predstavuje základný komunikačný protokol siete na výmenu dátových paketov a prepínaní protokolov medzi dvoma zariadeniami v sieti

Prehľad dostupných opcíí	
Výrobca	Názov opcie
Kuka	Kuka RSI
ABB	ABB EGM (nahradza ABB RRI)
Stäubli	Alter

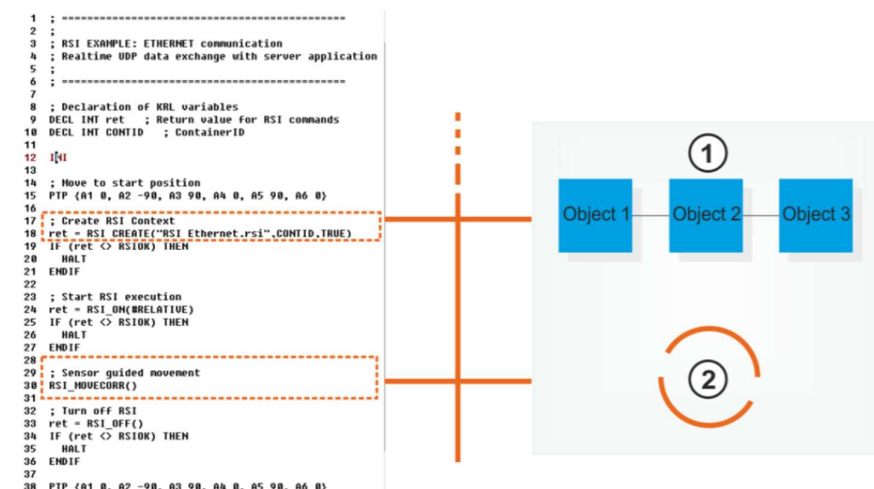
Tabuľka 1 - prehľad dostupných opcíí

2.2. Opcia od výrobcu KUKA - KUKA RSI

2.2.1. Úvod

Kuka RSI (**R**obot **S**ensor **I**nterface) je rozhranie robotov Kuka pre komunikáciu medzi robotom, respektíve jeho PLC a externými senzormi, teda PLC na ktoré sú roboty pripojené. Dáta sú prenášaná cez ethernet alebo I/O porty priamo do robota, pričom signál je spracovávaný rovnako rýchlo, ako je cyklus impulzov signálov zo snímača. Dáta zo snímačov priamo vplývajú na pohyby robota a je nimi možné priamo korigovať dráhu robota počas vykonávanie funkcie, Senzor disponuje maximálnymi možnými hodnotami, ktoré vracia do systému, a je teda možné čiastočne odhaliť, ak by prišlo k jeho poruche a vracal by chybné hodnoty Cez RSI monitor je možná on-line vizualizácie prenášaných dát zo senzora. Pre komunikáciu s robotom nie je špecifikovaný protokol prenášaných dát, je nutné aby si to špecifikoval sám užívateľ v súbore formátu XML. Tie dáta sú real-time, čo je nevyhnuté pre priame korekcie robota. Na vstupe do Kuka RSI sú priamo dáta zo snímača, na jeho výstupe sú korekcie pre robota. To znamená, že RSI funguje v podstate ako samostatné PLC ktoré spracováva program pre robota, aj keď je integrované v robotovi. Priame spracovanie dát zo senzoru môže byť vypnuté alebo zapnuté cez KRL, a v závislosti na použitom móde je rýchlosť spracovanie dát zo snímača 12ms/4ms (PO mode/PO_FAST mode). RSI umožňuje korigovať buď priamo uhol natočenia osy, v prípade že ide o senzor

snímajúci tento uhol, prípadne pokiaľ senzor sníma súradnice v karteziánskom súradnicovom systéme, je možné robiť korekcie priamo v karteziánskom súradnicovom systéme.



Obrázok 2 - Výmena dát vrámci RSI [1]

2.2.2. Základné termíny používané výrobcom KUKA

CLI – KUKA Line Interface – zbernica pre integráciu systému (robota) do siete zákazníka

KRC – Kuka Robot Controller – Riadiaca a ovládacia časť robota

RSI container – predstavuje objekt vytvorený v KRL programe ktorý obsahuje tok signálov definovaný v RSI Visual. Tento tok sa označuje aj RSI context a pozostáva z RSI objektov a väzbami (linkami) medzi jednotlivými objektami

RSI container ID – jedinečné identifikačné číslo priradené každému containeru

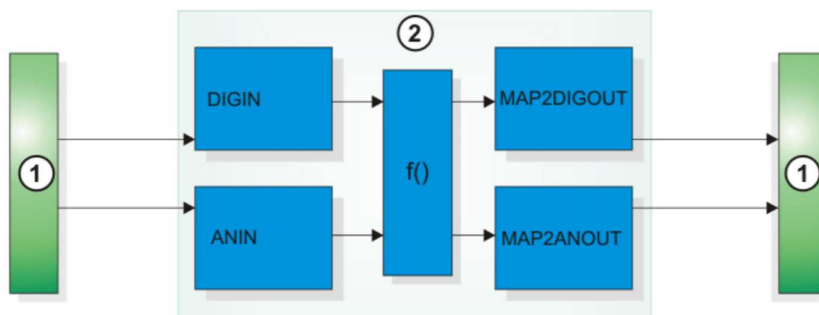
RSI Monitor – funkcia na monitorovanie a zobrazovanie RSI signálov (popis je uvedený neskôr v práci)

RSI Knihovna – knihovňa ktorá obsahuje všetky RSI objekty ktoré sú dostupné pre signálový tok v RSI Visual. Tieto objekty slúžia na správnu konfiguráciu signálneho toku a majú definované parametre, podľa ktorých vracajú návratovú hodnotu (Return Value)

RSI Visual – Grafický editor na spracovávanie signálneho toku, prácu s objektami a ich linkovanie medzi sebou

2.2.3. Vstupy a výstupy cez I/O systém

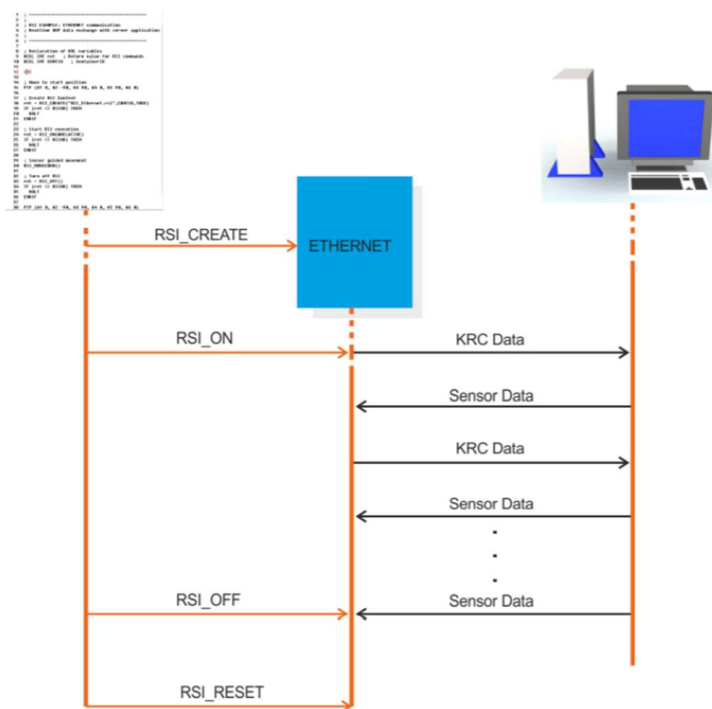
Pre signály vystupujúce zo snímača môžu byť použité buď DIGIN alebo ANIN, podľa toho či sa jedná o analógový alebo digitálny signál. Pre výstup z Kuka RSI je možné použiť MAP2DIGIT OUT, respektíve MAP2ANOUT, podľa toho, i snímač alebo PLC ktoré spracováva jeho signály komunikuje cez analógový alebo digitálny signál.



Obrázok 3 - Výmena dát cez A/D I/O porty [1]

2.2.4. Vstupy a výstupy cez ETHERNET

Pri komunikácii cez ethernet port je nutné mať na pamäti, že takáto komunikácia len emuluje priame porty. To znamená, že reálne pripojený 1 port sa správa ako 64 vstupných a 94 výstupných signálov. Táto emulácia je daná systémovo a teda nie je možné ju nejako narušiť alebo pozmeniť. Pre programátora to teda zjednodušené znamená, že pripojením jedného ethernet portu si vytvorí 128 či už analógových alebo digitálnych portov, s ktorými má možnosť pracovať v Kuka RSI. Každý jeden port sa správa ako samostatná jednotka, bez ohľadu na to, či sú pripojené cez jeden ethernet port. Komunikácie po ethernet porte prebieha cez UDP/IP protokol. Je však nutné mať na pamäti, že táto komunikácia prebieha iba na úrovni ethernet portu, a pri spracovaní na úrovni Kuka RSI je nutné špecifikovať protokol prenášaných dát cez protokol uložený v XML file, ktorý musí byť uložený v ETHERNET object-e.



Obrázok 4 - komunikácia cez Ethernet [1]

2.2.5. Real-Time odpoveď

Pre zaručenie, že dáta zo senzoru budú aktualizované real time a nebude dochádzať k oneskoreniu prijímaných dát, musí byť splnená podmienka, že každý packet sa zo senzoru (teda respektíve že časový rozdiel medzi dátovými prichádzajúcimi zo senzoru) nepríde v časovom rozostupe väčšom ako je dovolený v real-time móde robota (12 ms PO mode/4 ms PO_FAST mode). Pri presiahnutí tohto času dôjde ku zastaveniu robota (týmto je garantované, že dáta zo snímača sú naozaj real-time a nedochádza k ich výraznému oneskoreniu, a teda k nabúraniu robota z dôvodu zlých/oneskorených dát zo snímača). Avšak je treba mať na pamäti, že v tomto móde je to senzor, ktorý robí priame korekcie dráhy a nie robot ako taký, a teda za mechanické poškodenie robota na základe zlých dát zo senzoru zodpovedá užívateľ, respektíve technik inštalujúci senzor.

2.2.6. Možnosti korekcie

Dátami zo snímača je možné korigovať polohu priamo osí robota, ale tak isto aj externých osí. Pre korekciu osí robota je nutné použiť príkaz AXISCORR, pre korekcie externých osí príkaz AXISCORREXT. Korekcie je možné primárne robiť v karteziánskom súradnicovom systéme, ale sadou príkazov je možné túto korekciu previesť do všetkých rôznych súradnicových systémov, s ktorými môže robot pracovať. (BASE, ROBROOT, TOOL, WORLD, Tool,based)

Pre korekciu dráhy robota je možné tak isto použiť príkaz RSI_MOVECORR(), kedy je robot od začiatku pohybu smerovaný nie pomocou endpointu, ale na základe dát zo snímača.

2.2.7. Safety - bezpečnosť

Je nutné mať na pamäti, že v prípade zasahovanie do dráhy robota cez Kuka RSI je možné prísť k ujme na zdraví alebo majetku chybou spôsobenou na snímači. V tomto prípade je zodpovednou osobou nastavovač, a je nutné mať nastavené softwarové limity tak, aby nedošlo k majetkovej alebo zdravotnej ujme. Softwarové limity môžu byť jedným z prvkov, vďaka ktorým je možné odhaliť chybu snímača/snímačov. Nastavovač by mal nastaviť softwarové limity tak, aby boli minimálne možné pre splnenie pohybových inštrukcií, teda aby umožňovali bezproblémový chod robota, ale aby zároveň vylúčili riziko ujmy na majetku alebo zdraví. Softwarové limity je možné nastaviť karteziánske, ale zároveň aj na pohybové limity jednotlivých osí. Senzor však môže upravovať dráhu robota v rozsahu ± 5 mm alebo $\pm 5^\circ$ v závislosti od toho, či je použitá korekcia na základe karteziánskych súradníc, alebo dát o natočení jednotlivých osí, celkovo je však možné dospieť až ku korekciám ± 6 mm alebo $\pm 6^\circ$.

2.2.8. Systémové požiadavky pre využitie Kuka RSI

Pre používanie KUKA RSI je nutné disponovať kontrolérom rady KR C4 alebo vyšším, pripojením robota pomocou ethernet alebo profinet (pre využitie I/O vstupov) a zároveň externým PC (PLC) pre možnosť zobrazovania a upravovania dát medzi senzorom a robotom., z hľadiska software je nutné používať Kuka Software 8.3 a vyššie a zároveň operačný systém Windows disponujúci technológiou .Net Framework 3.5. Úpravy dráhy pomocou Kuka RSI sú garantované iba na 6- osích manipulátoroch, pre použitie na iných Kuka manipulátoroch je nutná konzultácia priamo s výrobcom Kuka. Pri

použití KUKA.Roboteam je nutné zaručit, aby nedocházalo k ovplyvňovaniu súradníc „Slave“ robota korekciami „Master“ robota.

2.2.9. Inštalacia, aktualizácia a odinštalovanie RSI do manipulátoru, inštalácia RSIVisual na externé PC

Samotný postup inštalácie je v oboch prípadoch veľmi intuitívny. Treba mať na pamäti, že v prípade robota sú požadované najvyššie „EXPERT“ prístupové oprávnenia, v prípade externého PC zase administrátorské práva. Pri prenose dát na Robot je veľmi odporúčané použiť priamo KUKA USB Data stick, s ohľadom na zachovanie dát na nosiči a jeho kompatibilitu. Pri zasahovaní do softwaru je vhodné zálohovať všetky dáta.

2.2.10. Konfigurácia sieťového spojenia

Pre komunikáciu cez sieť vyžaduje RSI modul vlastné ethernetové pripojenie nezávislé na inej KLI podsieti. Pre túto komunikáciu je možné využiť buď slot s označením X66 alebo X67.1-3.

2.2.11. Nastavenie RSI modulu

Pre nastavenie RSI modulu sa používa RSIVisual na externom PC. Tento program je dostupný buď v nemčine alebo angličtine. Pre prácu je používaný signal flow editor, kde je možné linkovať jednotlivé signály, nastavovať vlastnosti jednotlivým objektom a upravovať jednotlivé prístupové práva. Disponuje aj funkciou RSI monitor, kde má užívateľ možnosť zobrazovať a nahrávať 24 signálov.

2.2.12. Základný prehľad KUKA RSI príkazov

RSI_CREATE() – Tento príkaz vyvolá vytvorenie RSI container-u a načíta doňho signálový tok (Signal flow) z RSI VISUAL. Prístupovým parametrom na container je jeho ID, po jeho vytvorení je automaticky aktivovaný.

RSI_DELETE()- Táto funkcia slúži na odstránenie RSI containeru a jeho súčastí

RSI_ON()- Táto funkcia aktivuje spracovávanie signálov a zadefinuje korekcie a snímacie parametre senzorov. Pri vytvorení nového containeru dôjde k automatickému vykonaniu procedúr v tomto príkaze.

RSI_OFF()- Pri tomto príkaze dôjde k deaktivácii spracovávania signálov

RSI_MOVECORR()- táto funkcia spustí senzorom riadený pohyb (teda jeho korekcie, buď priamo korekcie osí alebo polohy TCP). K deaktivácii príkazu môže dôjsť aj za pomoci RSI objektu STOP

RSI_GETPUBLICPAR()- Príkaz na získanie public parametru (za predpokladu že je parameter povolený v hlavičke RSI kódu)

RSI_SETPUBLICPAR()- Zápis do public parametru (znova platí že parameter je povolený v hlavičke RSI kódu)

RSI_RESET()- týmto príkazom dôjde k vymazaniu containerov, a ich obsahov a RSI objektov

RSI_CHECKID()- táto funkcia, zistí, či je ID containeru správne, teda vráti True, ak pre zadané ID v jeho parametre existuje container, pokiaľ nie tak vracia False

RSI_ENABLE()- týmto príkazom dôjde k aktivácii containeru, ktorého ID je v jej parametre

RSI_DISABLE() – funkcia deaktivuje container, ktorého ID je v jej parametre [1]

2.3. Opcia od výrobcu Stäubli – Stäubli Alter

2.3.1. Úvod

Technickým riešením od konkurenčnej firmy Stäubli, ktoré umožňuje real-time korekciu dráhy sa nazýva ALTER. Na rozdielnej od KUKA RSI podporuje Stäubli len korekcie v karteziánskej sústave. Nie je teda možné aplikovať priamo korekcie na natočenia alebo posunutie jednotlivých osí (externým enkodérom alebo pravítkom). Stäubli umožňuje korekciu priamo pre jednotlivé geometrické dráhy (MoveL, MoveJ a MoveC, teda pohyb po kružnici). Je teda možné priamo senzorom korigovať konkrétnu pohybovú inštrukciu. Je nutné však mať na pamäti, že aby bolo možné dáta zo senzoru Real-time, musí byť odozva PLC rýchla.

Pri programovaní korigovaných dráh je nutné najskôr nadefinovať základnú pohybovú inštrukciu, a až následne jej korekciu. Príkazový syntax je rovnaký ako pre nekorigované dráhy, používajú sa príkazy alterMoveL(), alterMoveJ() a alterMoveC(). Korigovanú dráhu si môžeme zdefinovať ako jeden až viacej za sebou nasledujúcich príkazov s korekciou medzi dvoma príkazmi bez korekcie.

Samotná korekcia je aplikovaná príkazom alter(). Podľa typu dráhy je možné použiť variabilné módy korekcie, ktoré sa začínajú príkazom alterBegin(). Príkaz AlterEnd() je vyžadovaný najmä preto, aby bolo jasné ako má byť korekcia ukončená. Tento príkaz musí byť vykonaný ešte predtým, ako sa korigovaný pohyb skončí. Toto je nutné z dôvodu, aby bol prechod na následný nekorigovaný pohyb plynulý a bez zastavenia. Pokiaľ je tento príkaz až po ukončení dráhy, dochádza pohybu po nekorigovanej dráhe s poslednou známou korekciou z korigovanej dráhy, až pokiaľ nedôjde k zastaveniu.

Z dôvodu že ku korekciám dochádza prakticky okamžite, je nutné aby bola korekcia a dáta so snímačov kontrolované, aby nedošlo k nespojitým pohybom alebo vibráciám. Veľké zmeny korekcií sú možné iba so špeciálnou kontrolou, aby bola prípadne ošetrovaná náhla porucha senzorov. Pri zastavení korekcií je nutné, aby bola zmena korekcií nulová, prípadne nulová pohybová rýchlosť korigovaného pohybu. Táto možnosť je tiež systémovo ošetrovaná špeciálnou kontrolou. Pri synchronných príkazoch dochádza k odosielaní dát o pozícii a rýchlosti do zosilňovača. V dôsledku tohto musia byť korekčné príkazy zladené s touto komunikačnou periódou, aby bola korigovaná rýchlosť kontrolovateľná. Toto možné pomocou špeciálneho synchronizačného príkazu. Rovnakým spôsobom môže najskôr odfiltrovaný vstup zo snímača, pokiaľ sú dáta rušené alebo pokiaľ je perióda odozvy snímača iná ako je kontrolná perióda kontroléru.



Obrázok 5 - 6-osý manipulátor Stäubli [2]

Pokiaľ je vyžadovaný hladký prechod medzi jednotlivými korigovanými a následnými nekorigovanými pohybmi, je nutné dostatočne včas pre ukončením pohybu spustiť alterEnd príkaz. Pokiaľ dôjde k jeho spusteniu blízko koncu pohybu, je možné že dôjde ku spomaleniu alebo až zastaveniu pohybu, aby bolo možné dosiahnuť počiatočný bod novej dráhy. Navyše, možnosti dosiahnutia počiatočného bodu v predstihu majú isté obmedzenia na korigovanej dráhe po tom ako bol vyvolaný príkaz alterEND. Napríklad už potom nie je možné aby došlo ku zmene konfigurácie a nie je možné zaručiť, že sú všetky osy v správnej orientácii. Tým pádom môže dôjsť ku chybe, napríklad konfigurácie, ku ktorej by normálne nedošlo, pokiaľ by bol príkaz alterEnd zavolaný s dostatočným predstihom.

2.3.2. Safety - Bezpečnosť

Užívateľská korekcia môže byť neplatná pokiaľ nastanú prípady ako bod mimo dosahu, rýchlosť alebo akcelerácia vyššia ako systémové limity. Pokiaľ takýto stav nastane, dochádza k vygenerovaniu chyby a manipulátor je zastavený na poslednej platnej pozícii. Pre pokračovanie je nutné pohyb vyresetovať. Pokiaľ bol manipulátor zastavený počas pohybu (režim pozastavenia, požiadavka na zastavenie alebo Emergency stop), platia pre režim zastavenia (brzdné časy, rampy a rýchlosti) rovnaké podmienky ako pri bežnom zastavení. Po určitom časovom úseku dochádza aj k automatickému zakázaniu korekcií, aby bolo garantované úplné zastavenie manipulátoru. Pokiaľ je podmienka zastavenia anulovaná, pohyb môže pokračovať a korekcie sú znova automaticky povolené.

2.3.3. Obmedzenia

Nulový pohyb (teda keď je cieľový bod totožný so štartovým) je automaticky odignorovaný. Dôsledkom toho je, že pre začiatok korigovanej dráhy je vyžadovaný nenulový pohyb. Postačujúca je vzdialenosť 0,001 mm medzi počiatočným a koncovým bodom.

Rovnako nie je možné užívateľom meniť konfiguráciu pre korigovanú dráhu, systém automaticky používa rovnakú konfiguráciu. Preto nie je možná zmena konfigurácie manipulátoru v korigovanej dráhe (ani v prípade MoveJ inštrukcií)

Nie je ani možné resetovať korekciu pomocou príkazovej sekvencie alterEnd() a alterBegin() bez toho, aby došlo k vykonaniu príkazu movel, movej, movec alebo ResetMotion() medzi tým. Každá spojená korigovaná dráha podporuje iba jeden alterBegin() príkaz.

2.3.4. Základné robotické príkazy pre korekciu dráhy

alterMovej() – príkaz sa správa rovnako ako Movej(), akurát že umožňuje korekcie

alterMovel() - príkaz sa správa rovnako ako Movel(), akurát že umožňuje korekcie

alterMovec() - príkaz sa správa rovnako ako Movec(), akurát že umožňuje korekcie

alterBegin() – tento príkaz inicializuje mód korekcií

alterEnd() – tento príkaz ukončuje mód korekcií a prebiehajúci pohyb už nie je možné korigovať

alter() – tento príkaz začína a špecifikuje nové korekcie

alterStopTime() – tento príkaz zobrazí zostávajúci čas do ukončenia korigovanej dráhy

Presné informácie, typy premenných a prípadne význam return values sú uvedené priamo v manuáli od výrobcu. [3]

2.4. Opcia od výrobcu ABB – ABB Externally Guided Motion

2.4.1. Úvod

V prípade výrobcu ABB je nutné na online korekciu dráhy doplatiť si opcii Externally Guided Motion. Táto opcia umožňuje tri všeobecné funkcie, EGM Position Stream, EGM Position Guidance, EGM Path Correction. Pre online korekciu dráhy je vhodná funkcionálna EGM Path Correction. Funkcia EGM Path Correction je obmedzená iba na 6-osé roboty a umožňuje užívateľovi korigovať dopredu naprogramovanú dráhu podľa dát z externého snímača. Vyžaduje, aby počiatočný a koncový bod boli bez plynulého prechodu do nasledujúcej dráhy, teda definované ako fine. Senzor musí byť nainštalovaný na robotovi, a korekcie sú možné iba v súradnicovom systéme dráhy v osách Z a Y, teda v osách kolmých na smer (alebo tangentu) dráhy. V prípade kolízie je vhodné EGM vypnúť, opraviť chybu, prípadne sa vrátiť, a až následne znova aktivovať EGM. Snímacia sekvencia senzoru môže byť až do 48 ms. EGM je schopné korekcie aplikovať v rôznych framoch, a to konkrétne v Base, Tool, ktoré však ešte musia byť spolu vzájomne previazané. Pre definíciu každého je nutné použiť príkaz EGMMoveL alebo EGMMoveC podľa toho, v ktorom frame je daný príkaz zadaný. V tomto smere sú korekcie dráhy u ABB značne obmedzené.



Obrázok 6 - 6-osý manipulátor ABB [4]

2.4.2. Základný postup pri vytváraní korigovanej dráhy EGM

1. Násť do východzieho bodu (definovaný ako fine)
2. Definovať EGM klienta a EGM identitu (EGM stav bude zobrazovaný ako EGM_STATE_DISCONNECTED)
3. Zavolať EGM setup inštrukciu pre nastavenie dát o pozícií a definovaní vstupných signálov (EGM stav sa zmení na EGM_STATE_CONNECTED)
4. Definovať correction frame senzoru, ktorý je vždy zhodný s nástrojovým súradnicovým systémom
5. Následne bude prebiehať samotný pohyb (EGM stav bude EGM_STATE_RUNNING). V nasledujúcom bode trajektórie definovanom ako fine dôjde ku zmene stavu EGM naspäť na EGM_STATE_CONNECTED)
6. Pre použitie dát z iného senzoru je nutné resetovať EGM (v tomto prípade sa zmení EGM stav na EGM_STATE_DISCONNECTED)

2.4.3. Základné stavy EGM a ich význam pre korekciu dráhy

EGM_STATE_DISCONNECTED – Táto hláška znamená, že systém EGM nebol zadefinovaný a teda nie je možné aktivovať

EGM_STATE_CONNECTED – Znamená, že EGM systém je alebo bol úspešne zadefinovaný, ale nedochádza ku korekcií, keďže systém nebol aktivovaný.

EGM_STATE_RUNNING _ Systém bol úspešne zadefinovaný a dochádza ku korekcií dráhy.

Medzi poslednými dvoma stavmi existuje kauzalita, a robot sa môže samostatne prepnúť do stavu EGM_STATE RUNNING zo stavu EGM_STATE_CONNECTED, po príkaze ktorý mu definuje že má začať korigovanú dráhu. Tak isto je možné, že sa robot prepne zo stavu EGM_STATE_RUNNING naspäť na EGM_STATE_CONNECTED, pokiaľ robot dosiahne cieľového bodu trajektórie. Tak isto, pokiaľ dôjde k prerušeniu pokiaľ je robot v stave EGM_STATE_RUNNING, robot sa po odstránení chyby a znova aktivovaní vráti do stavu EGM_STATE_RUNNING.

Pokiaľ by sme toto nechceli využiť, je nutné použiť príkaz Do Not Restart after Motors Off, tento príkaz, pokiaľ je jeho hodnota rovná TRUE, vráti systém do stavu EGM_STATE_CONNECTED. Pokiaľ je jeho ale jeho hodnota FALSE, tak ako je nastavená jeho základná hodnota, robot sa vracia do stavu EGM_STATE_RUNNING, a teda začnú byť hodnoty z dát o korekciách znova aplikované. Povolené hodnoty pre príkaz Do Not Restart after Motors Off sú typu BOOL, teda TRUE alebo FALSE.

Ak by sme chceli, aby došlo po zastavení robota počas korigovanej dráhy, aby došlo k jeho vráteniu vopred nastavenej pozície, napríklad Home pozície, je nutné použiť príkaz Return to Program Position when Stopped. Táto hodnota je primárne nastavená na stav False, pokiaľ ju prenastavíme na stav True, dôjde k presunu robota na požadovanú pozíciu. V tomto prípade je nutné, aby nebola požadovaná nastavená pozícia mimo safety limitov, tak aby bol možný presun robota do tejto pozície. Inak Bude pohyb robota zastavený zo bezpečnostných obmedzení. Takže odporúčaním výrobcu je, aby požadovaná hodnota nebola príliš vzdialená. Pre príklad uvediem proces adaptívneho zvárania, kedy pri prerušení dôjde k odsunutiu hlavice o pár centimetrov a pri opätovnom spustení programu zvárania sa hlavica naspäť priblíži ku zvaru. Povolené hodnoty príkazu Return to Program Position when Stopped sú TRUE alebo FALSE.

2.4.4. Možnosti komunikácie s EGM

Pre komunikáciu s EGM robota je možné použiť analógový vstup, analógový výstup, skupinu analógových výstupov a tiež UdpUc (User Datagram Protocol Unicast Communication) protokol. Všetky z týchto vstupných signálov by mali obsahovať dáta o korekciách. Tieto údaje sú čítané v časovej slučke 48 ms alebo jej násobkoch. Podľa nich si EGM dopočítava hodnoty korekcií.

Komunikácia cez UdpUc - cez toto rozhranie je robot schopný komunikovať až s ôsmimi zariadeniami naraz. Zariadenie sa v tomto prípade javí ako server a robot ako klient, kde práve kontrolér robota je ten, ktorý inicializuje a nadväzuje komunikáciu. Pre nadviazanie komunikácie je nutné zadať názov protokolu, jeho typ, v prípade UdpUc je to UDPUC, meno seriálového portu (stačí číslo), pokiaľ by bola komunikácia zadefinovaná cez IP protokol, je typ nutné uviesť pred názov portu a tak isto aj dodatočne uviesť jeho vzdialenú IP adresu a vzdialené IP číslo portu.

2.4.5. Základné príkazy EGM

Príkaz EGMActMove slúži na označenie začiatku korigovanej dráhy. Iba použitím tohto príkazu môžeme trvalo definovať korekciu, napríklad ak si externý senzor nameria že obrobok je o 10 mm nižšie ako by mal byť, a teda posunie všetky body. Ide teda o najzákladnejší príkaz.

Ďalším z príkazov je EGMMoveC, ktorý definuje kruhový pohyb po zadanej kruhovej trajektórii (platia tie isté pravidlá ako v prípade príkazu MoveC). Pri tomto príkaze je vhodné robiť korekcie iba v ose Z, korekcie v ose Y alebo korekcie stredu kruhového pohybu sú veľmi náročné.

Príkaz EGMMoveL definuje korekcie v osách Z a Y (korekcie v ose X, teda v smere pohybu, nie sú z princípu u opcie EGM ABB možné). Ostatné parametre a použitie je

rovnaké ako v prípade Príkazu MoveL. Keďže príkaz MoveL je z hľadiska orientácie nástroja stacionárny, je možné robiť aj korekcie natočenia nástroja.

Tak ako som už niekoľkokrát písal, korekcie natočenia osí u ABB nie sú možné, a teda príkaz typu EGMMoveJ ani neexistuje. Toto je asi najväčšia škoda a chyba u korekcií u ABB z môjho pohľadu.

Príkaz EGMRreset slúži na reštart korigovaného pohybu. Či už sa jedná o príkaz EGMMoveC, EGMMoveL alebo EGMActMove, príkaz sa vráti na začiatok a začne odznova. Pokiaľ by sme tento príkaz využili v prípade prerušenia pokiaľ skenujeme obrobok, môžeme pôvodné dáta pred prerušením vymazať a nanovo naskenovať.

Príkaz EGMSetupAI slúži na zadefinovanie analógových vstupov. Pre základné použitie stačí zadefinovať číslo analógového vstupu, pri väčšej náročnosti je možné nastaviť rôzne parametre vstupov, napríklad o korekcie akého pohybu sa bude daný vstup starať a ďalšie iné.

Príkaz EGMSetupAO slúži na zadefinovanie analógových výstupov, tak isto ako v predchádzajúcom prípade stačí nadefinovať číslo výstupného portu, pokiaľ má užívateľ záujem, je možné definovať aj rôzne ďalšie parametre.

Príkaz EGMSetupGI sa stará o definíciu viacerých vstupov, ktoré bude EGM využívať. Rovnako ako v prípade príkaz EGMSetupAI je možné zadefinovať v základe iba čísla portov, cez ktoré budeme komunikovať. Pokiaľ je užívateľ náročnejší, môže si zadefinovať viaceré parametre vstupných hodnôt, napríklad ktoré porty budú používané pre zadané príkazy a iné.

Príkaz EGMSetupLTAPP slúži na definíciu LTAPP protokolu rozhrania pre komunikáciu so senzormi. Aktuálne sa výrobca ABB pozerá na toto rozhranie ako na budúcnosť viacerých analógových vstupov a toto rozhranie má relatívne malá využitie. Aj tak podporuje všetky rôzne funkcie ako príkazy EGMSetupAI, EGMSetupAO a EGMSetupGI.

Pomocou príkazu EGMSetupUc je možné definovať komunikáciu cez UdpUc protokol. Popis tejto komunikácie so senzormi bol už opísaný vyššie a je možné ním pridať až 6 doplnkových portov pre korekcie, čo predstavuje značne rozšírené možnosti, pokiaľ sú na robotovi použité externé osy a dochádza k pohybu v týchto osiach (napríklad ak má robot robiť manipulácie na dopravníku ktorý sa hýbe vopred nedefinovanou alebo rôznou rýchlosťou).

A na záver príkaz EGMStop, ktorý slúži na ukončenie všetkých Externally Guided Motions, teda na návrat k základným pohybom. Tento príkaz sa objavuje najmä pri prechode z korigovanej dráhy na dráhu nekorigovanú, pokiaľ je vyžadovaná rýchla zmena medzi nimi. [5]

2.4.6. Zhodnotenie vlastností EGM

Celkovo má opcia EGM od výrobcu ABB veľmi široké použitie a umožňuje užívateľovi množstvo dodatočných funkcií. V prípade korekcie dráhy však nie je taká rozsiahla ako opcia RSI od výrobcu KUKA, ktorá je priamejšie mierená na korekcie dráh podľa dát zo snímačov. Opcia od ABB je však širšia a teda je jej flexibilita v použití vo výrobe

a pri zmene výrobného procesu väčšia, a teda aj roboty s dokúpenou touto opciou sú lepšie použiteľné. Z hľadiska programovania korekcie dráhy však nie sú tak programátorsky prívetivé ako tie od výrobcu KUKA, ktoré majú dokúpenú opciu KUKA a ich programovanie cez KUKA RSI a najmä s použitím RSI Monitoru je výrazne lepšie a najmä vhodnejšie pre odlaďovanie chodu a prípadných chýb v programe.

3. Porovnanie jednotlivých opcí a zdôvodnenie výberu zvoleného riešenia

Základným rozdielom medzi opciami od jednotlivých výrobcov je ich obšírnosť. Zatiaľ čo opcia od výrobcu ABB je najširšia, opcia od Stäubli je najjednoduchšia. Rovnako je to aj s náročnosťou na programovanie. Veľkou výhodou opcie od výrobcu KUKA sú dostupné balíky programov VisualRSI a RSI Monitor, ktoré poskytnú programátorovi dôležité opory pri odlaďovaní programu. Naopak výhodou balíkov od výrobcov ABB a Stäubli je diferenciacia príkazov pre korigované a nekorigované dráhy, čo umožňuje programátorovi prekladať jednotlivé korigované a nekorigované dráhy. Opcia od výrobcu ABB je navyše ešte veľmi obšírna a umožňuje množstvo ďalších funkcií okrem korekcií dráhy.

Pri výbere však najväčšiu úlohu zohráva dostupný robot, prípadne aké roboty sú na hale alebo od akého výrobcu fabrika nakupuje celé dodávky. Osobne mi pre korekcie dráhy príde najlepšia opcia KUKA RSI.

Aj v mojom prípade však nebola možná voľba ale mal som použiť priamo robota KUKA KR16, ktorý už bol osadený v dielni aj so senzorom od Keyence. Preto som praktickú časť mojej práce spracovával práve na ňom.

4. Aplikácia korekcií na reálnom pracovisku

4.1. Úvod – zoznámenie s pracoviskom

Moju diplomovú prácu som realizoval na pracovisku v dielni C1 na našej fakulte. Na pracovisku sa nachádza 6-osý manipulátor Kuka KR16, na ktorom je namontovaná snímacia hlavica IL-300 a zosilňovač IL-1000. Hlavica sa nachádza priamo nad prírubou a zosilňovač na hornej časti manipulátoru. Na prírubu sa nachádza hrot, ktorý predstavuje v mojom prípade nástroj, ktorým jazdím po žiadanej dráhe. Tento je v smere ose z posunutý o 250 mm, v smere ose y o 80 mm. Natočením príruby, teda pootočením osy A6 je možné nastaviť, či bude lúč posunutý v smere osy y (pôjde teda paralelne s hrotom), alebo v smere osy x (a pôjde pred hrotom v určitom predstihu). Na robotovi je už nainštalovaná opcia RSI a tak isto je už aj pripojené PLC s riadiacim systémom BeckHoff. Robot je pripojený do siete s adresou 100.65.12.19, takže je možné sa naňho veľmi jednoducho pripojiť. Pre prácu s robotickým programom a jeho odladovanie a nahrávanie nových verzii som využíval program WorkVisual 6.0 od Kuky. Stačí sa počítačom cez ethernetový kábel pripojiť do siete a nastaviť IP adresu. Žiaľ nie je možné priamo pristupovať k C:\ disku robota, preto je nutné .xml súbory na korekciu nahrávať na zložku na disk D:\ a odtiaľ cez pendant prehrať na správne miesto na disku C:\. Pre prácu s xml súbormi na korekciu som používal program RSIVisual tak isto od Kuky. Tieto programy sú voľne dostupné. Prácu s obidvoma súbormi priblížim v ďalších podkapitolách. Cieľom práce je vytvoriť taký program na korekciu, aby pre prejazd ponad ľubovoľne tvarovanú dráhu postačovalo nastaviť počiatočný a koncový bod trajektórie, a všetky korekcie v smere osy Z boli automatizované podľa dát zo snímača vzdialenosti. Následne budú ešte údaje zo snímača a údaje o hodnote korekcie v ose Z a polohe v ose Z prenášané do PLC kde budú zobrazované na monitore. Na vytváranie programu pre PLC som použil program TwinCat, ktorý je doplnkom do programu VisualStudio od Microsoftu. Konkrétne som mal nainštalované VS 2017, s ktorým som pracoval už pred tým a ku ktorému som si len doinštaloval TwinCAT 3 od BeckHoff. V práci sa pokúsim robiť korekcie najskôr iba na základe sínusového signálu, aby som si overil či vôbec korekcie fungujú, následne sa pokúsim nastaviť korekcie na ľubovoľnú dráhu, kde snímaná vzdialenosť bude posunutá v ose Y (teda pôjde paralelne vo vzdialenosti 80 mm a na záver sa pokúsim vypracovať taký program na korekcie, aby bol snímaný bod posunutý v smere osy X, teda v smere pohybu robota. Korekcie budú teda musieť byť oneskorené.



Obrázok 7 -a) KUKA KR16, b) PLC na pracovisku KUKA UVSSR

4.1.1. Snímač Keyence IL-300

Laserová snímacia hlavica Keyence IL-300 využíva na snímanie vzdialenosti laserový lúč, vytváraný polovodičovou diódou o vlnovej dĺžke 655 nm. Jedná sa teda o červený lúč vo viditeľnom spektre. Priemer lúča je 0,5mm, dosahovaný výkon 560 μ W, a zariadenie je kategorizované ako 2 laserová trieda. Laser nespôsobuje výraznejšie ohriatie a nespôsobuje popáleniny na koži, nie je ho ani cítiť na koži. V mieste dopadu je viditeľný ako červený bod, takže je možné vidieť kde dochádza k odrazu lúča. Číslo v názve 300 označuje referenčnú snímáciu vzdialenosť 300 mm od snímacej hlavice. Od tejto referenčnej vzdialenosti sú krajné snímacie hodnoty vzdialené ± 140 mm, rozsah snímanej vzdialenosti je teda 160 až 440 mm. Hlavica je malá (66x50x27mm) a ľahká (135g), takže pri bežnom použití napríklad na koncovom efektore robota nespôsobuje veľké zmeny v momentoch. Opakovateľnosť laseru je 30 μ m. Pri montáži je potrebné hlavicu odizolovať podložkou. Pre zapojenie do zosilňovača Keyence IL-1000 je použitý 4-žilový tienený kábel so špeciálnym konektorom, ktorý je ku snímaču dodávaný. Hlavica využíva triangulačný princíp merania, ktorý nie je závislý na intenzite odrážaného lúča alebo čase, za aký sa odrazí. Tento princíp spočíva v tom, že pokiaľ

dochádza k odrazu lúča pod konštantným uhlom bez ohľadu na vzdialenosť, môže byť vzdialenosť meraného objektu zistená podľa toho, kam lúč dopadá. Pre lepšie vysvetlenie je lepšie si pozrieť obrázok. Presnosť hlavice teda nie je závislá na schopnosti merať intenzitu dopadajúceho lúča alebo časové intervaly, ale práve schopnosti rozlíšiť kam na dopadové pole, CCD senzor, odrazený lúč dopadá. Aj z tohto dôvodu má tento senzor stanovený rozsah, v ktorom meria. Ten je technologicky ohraničený veľkosťou dopadovej plochy, referenčná hranica je nastavená do stredu ale je možné si ju, ako aj v mojom prípade, ešte softwarovo posúvať. Toto už však nie je možné s rozsahom, ktorý je pevne daný práve vyššie uvedenými obmedzeniami. Z princípu funkcie je teda jasné, že snímač nemá citlivosť napríklad na farbu meraného predmetu, alebo aj na jeho lesk (avšak v praxi vykazuje vyššie kmitanie výstupného signálu). Je teda omnoho stabilnejší ako napríklad difúzny snímač. Navyiac je výhodou triangulačných snímacích hláv možnosť snímať aj cez priesvitné povrchy, potlačiť pozadie, popredie pri snímaní cez takéto povrchy. Tak isto je možné týmto typom snímačov merať aj hrúbku priesvitnej vrstvy. V takomto prípade sa na hlavici zobrazia dva dopadové lúče a teda vracia dve hodnoty. Avšak konkrétne hlavica IL-300 v kombinácii so zosilňovačom IL-1000 na takéto použitie nie je vhodná. [6]



4.1.2. Zosilňovač IL-1000

Samotný zosilňovač IL-1000 je kompatibilný s viacerými hlaviciami. Svojimi rozmermi už nie je úplne vhodný na montáž na koncový efektor, vhodnejšie je ho umiestiť na nepohyblivú časť robota alebo zariadenia. Je možné ho upevniť na DIN lištu. Tak isto je veľmi odporúčané zosilňovač umiestniť na dostupné miesto, pretože tlačidlami na ňom umiestnenými je možné nastavovať jeho vstupné a výstupné parametre. Tieto nastavenia je však možné robiť aj vzdialene cez kábel, čo je vhodné najmä pri častých automatizovaných zmenách alebo pri použití viacerých párov snímačov a zosilňovačov v jednej bunke. Zosilňovač je napájaný jednosmerným napätím, ktoré môže byť v rozsahu 10-30 V. Výstupom zo zosilňovača je 12 žilový vodič, žila pre analógový výstup je zatienená. Na zosilňovači je možné nastaviť si požadovaný analógový výstup, podľa toho na akú kartu (v našom prípade na robotovi) chceme zosilňovač pripojiť. Výstup môže byť buď napäťový v rozsahoch ± 5 V, 1-5 V, 0-5 V (výstupná impedancia je 100 Ω), alebo je možné použiť aj prúdový výstup v rozsahu 4-20 mA, pričom maximálny záťažový odpor môže byť 350 Ω . Zosilňovač má priamo na svojom tele

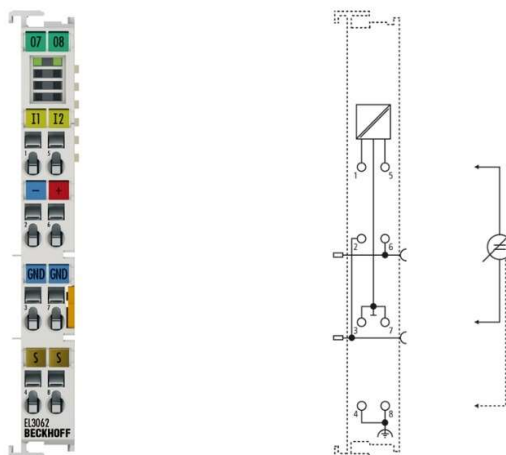
display a tlačidlá, ktorými je možné nastaviť požadovaný analógový výstup, a na displeji dochádza k zobrazovaniu nameranej hodnoty na desatiny mm. Toto je skôr myslené ako orientačný údaj, ktorý je možné využiť pri odlad'ovaní senzoru a pri prvom nastavení sú tieto údaje viac než vhodné. Keďže zosilňovač je kompatibilný s viacerými hlaviciami, je možné na ňom nastaviť v akej vzdialenosti je referenčná vzdialenosť hlavice a hornú a spodnú snímaciu hranicu. Tieto údaje však slúžia iba pre zobrazovanie na displeji, analógový výstup neovplyvňujú. [7]



Obrázok 9 a) zosilňovač IL-100 na KR16, b) detail na zosilňovač [7]

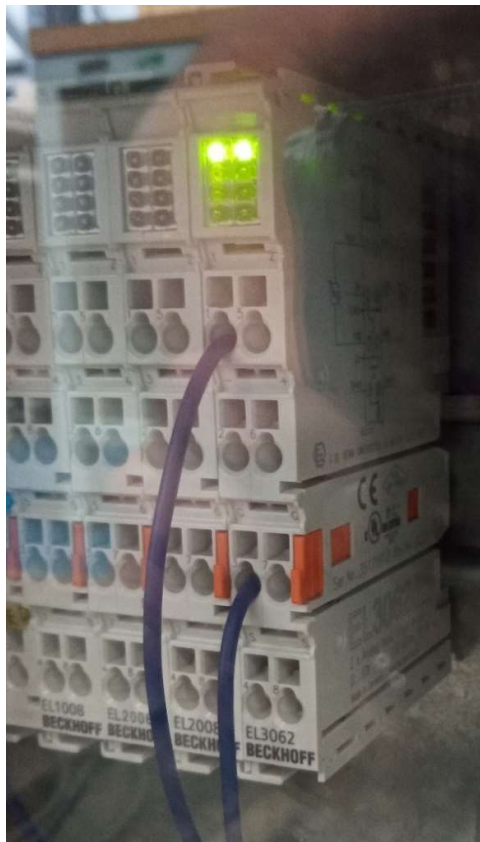
4.2. Zapojenie a sprevádzkovanie pracoviska

Manipulátor Kuka KR16 bol už na začiatku mojej práce pripojený do siete a opcia RSI už bola nainštalovaná s predchádzajúcich prác. Tak isto už bolo zasieťované PLC, v mojom prípade stačilo teda zapojiť senzor. Snímacia hlavica aj zosilňovač už boli namontované, hlavica bola zapojená do zosilňovača a bol z neho vyvedený viacžilový kábel. Bolo treba napojiť snímač na zdroj jednosmerného napätia 24V, ktorý bol v rozvodnici robota a pripojiť analógové výstupy na správnu kartu. V našom prípade vyhovovala osadená analógová vstupná karta Beckhoff EL3062, ktorá je zapojená do cupplera priamo na robota, s rozsahom 0-10V.



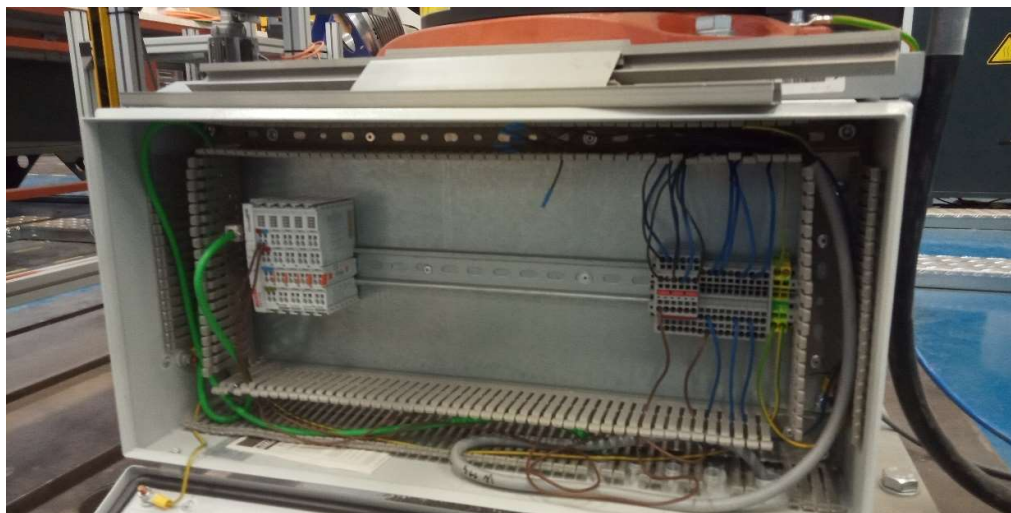
Obrázok 10 - karta Beckhoff EL3062 a schéma zapojenia [8]

Karta je napájaná priamo z cupplera. Snímač umožňuje, ako som už písal pri jeho popise, nastaviť výstupný rozsah analógového výstupu.



Obrázok 11 - EL3062 zapojená na pracovisku

V mojom prípade som nastavil rozsah 0-5V, ktorý najviac vyhovuje vstupnej karte. Karta je pripojená priamo do robota a hodnoty vstupu do nej je možné vidieť pod analógovým vstupom číslo 7. V tomto stave je už pracovisko pripravené a môžem začať riešiť problémy s korekciou dráhy.



Obrázok 12 - Cuppler umiestený pod KR16

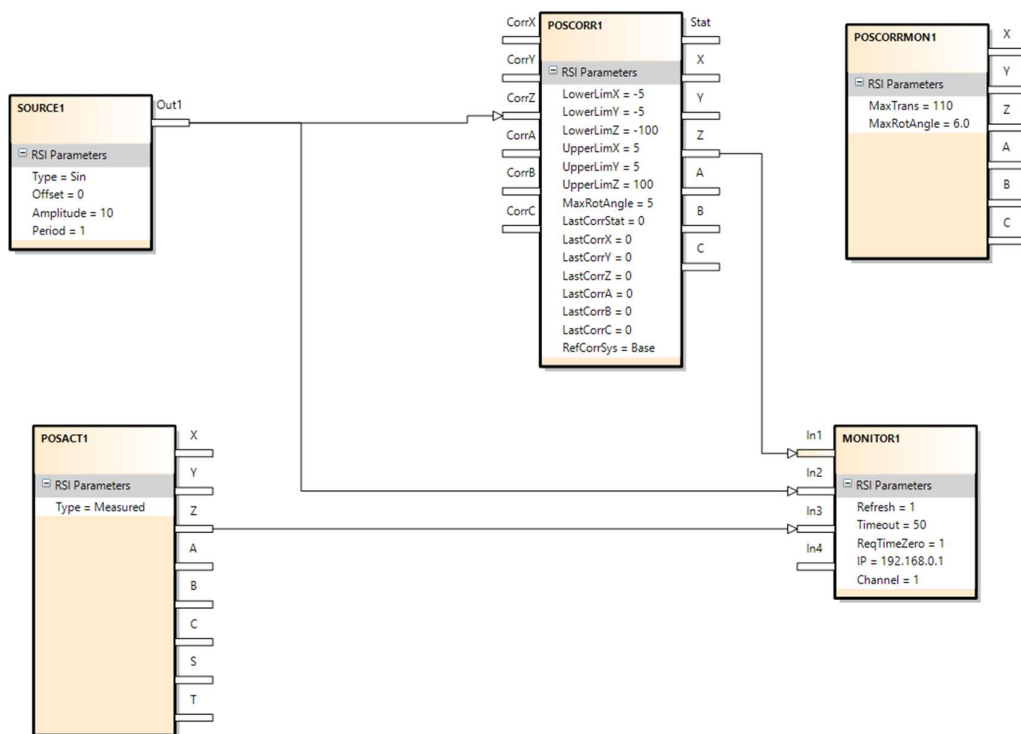
4.2.1. Nastavenie výstupného signálu zo snímača IL-1000

Pre nastavenie rozsahu výstupného signálu je nutné zosilňovač najskôr vyresetovať podržaním tlačidla MODE a SET po dobu 5s. na displeji sa objaví rESet, čím začína nové nastavenie zosilňovača. Na úvod je potrebné potvrdiť začiatok resetovania navolením voľby YES a potvrdením tlačítkom MODE. Následne je možné navoliť PNP alebo NPN výstup. V našom prípade volíme NPN a znova tlačítkom MODE pokračujeme na možnosti výstupu. Je možné nastaviť napäťové výstupy v rozsahoch 0-5 V, $\pm 5V$, 1-5V, alebo prúdový v rozsahu 4-20mA. Prípadne je možné výstup aj vypnúť. Potvrdíme tlačítkom MODE a parametre analógového výstupu sú nastavené. [9]

4.3. Vytvorenie a nahratie programu pre korekcie dráhy podľa sínusového signálu

4.3.1. Program v RSI

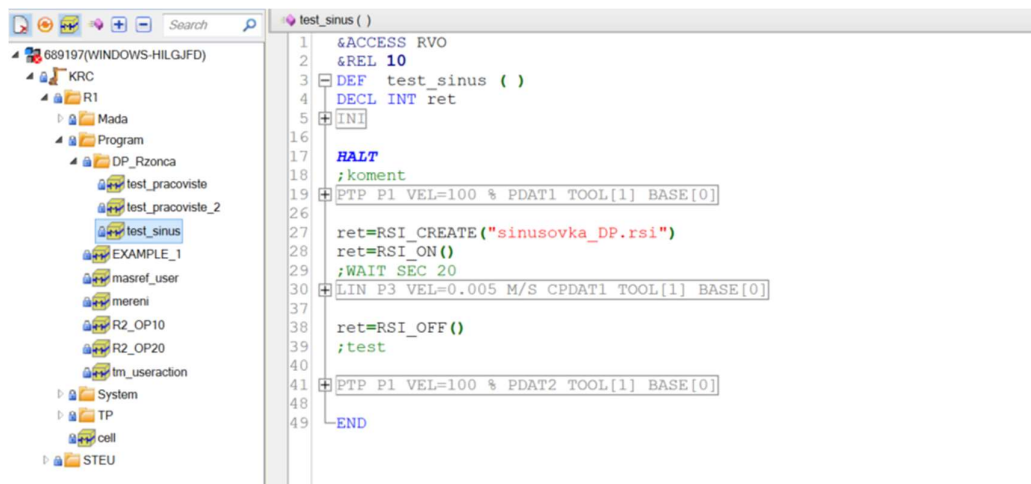
Prvým z programov, ktorý budem vytvárať, bude robiť korekcie lineárnej dráhy podľa sínusového signálu. Korekcie teda ešte nebudú podľa dát zo snímača, ale len podľa signálu generovaného v RSI. Tento program je vhodný na začiatok a pomôže mi lepšie pochopiť základné funkcionality RSI. Vytvorenie nového .xml súboru je veľmi jednoduché a vytvára sa pomocou blokovej schémy. V bočnej lište sa nachádza toolbox kde sú v rôznych kategóriách zaradené všetky dostupné bloky. Pre korekcie je potrebné použiť blok s názvom POSCORR, do ktorého je možné priviesť korekcie súradníc aj osí. V bloku je možné nastaviť limitné hodnoty korekcií (posunutia aj natočenia) a súradnicový systém ku ktorému sa budú korekcie vzťahovať. Ďalší z blokov, ktorý budeme potrebovať, je blok POSCORRMON, ktorý slúži na SEFETY funkciu. V tomto bloku nastavíme Maximálne rozsahy od pôvodnej dráhy alebo natočenia, o ktoré je možné korekcie robiť. Toto je veľmi vhodné, pokiaľ by došlo napríklad ku chybe snímača a manipulátor by dostal zo snímača chybné hodnoty. Blok POSACT slúži na zobrazenie aktuálnej pozície robota. Toto si samozrejme môže užívateľ pozrieť aj na touchpade, ale tento blok je možné použiť s funkcionalitou RSI Monitor a vyexportovať tieto údaje do .dat súboru. Pre využitie funkcie RSI monitoru je nutné vložiť ešte blok MONITOR, na ktorého vstupy si napojíme žiadané signály, ktoré si chceme zobrazovať. Sínusový signál budem generovať blokom SOURCE, kde nastavím typ výstupného signálu ako sínusový s amplitúdou 10 mm a periódou 1s. Výstup z bloku spojím so vstupom Z do bloku POSCORR. Nastavím v ňom rozsahy korekcií, bezpečnostné limity nastavím v bloku POSCORRMON. Do bloku monitor pripojím výstupy z generátoru signálu, výstup z bloku korekcií a výstup z bloku POSACT, ktorý mi poskytne informácie o aktuálnej polohe osy Z. Takto vytvorenú schému uloží, automaticky sa mi uložia 3 dokumenty vo formátoch .rsi, .rsi.diagram a .rsi.xml. Tieto tri súbory prekopírujem na C:\ disk do zložky C:\KRC\Roboter\Config\User\Common\SenzorInterface.



Obrázok 13 - ukážka programu pre korekcie na základe sínusového signálu

4.3.2. Robotický program

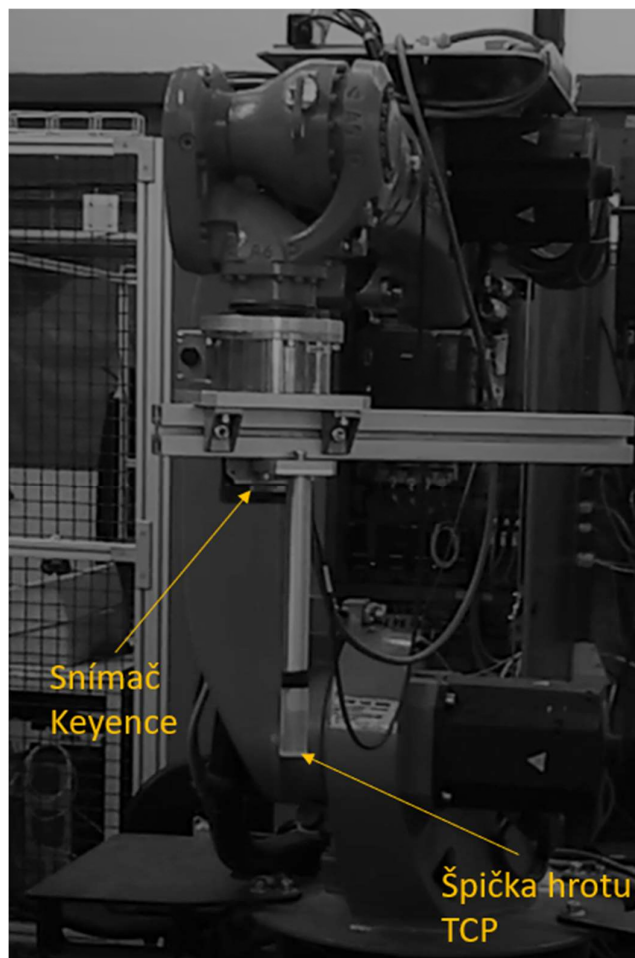
Následne je nutné vytvoriť robotický program, v mojom prípade som vytváral iba základný lineárny pohyb medzi dvomi bodmi. Prvým príkazom: `ret=RSI_CREATE("sinusovka_DP.rsi")` načítam údaje o korekcií. Časť `sinusovka_DP.rsi` predstavuje názov .rsi súborov, kde sú údaje. Je potrebné aby boli nakopírované všetky tri vygenerované súbory. Následne príkazom `ret=RSI_ON()` zapínam korigovanie dráhy pomocou dát z RSI. Po prejdení požadovanej dráhy príkazom `ret=RSI_OFF()` korekcie vypnem. Program som si pre zjednodušenie vytvoril v prostredí WorkVisual 6.0 a do robota iba zosynchronizoval cez sieť. Body som si našiel ručne cez TeachUp príkaz. Robot sa pohyboval medzi dvomi bodmi a pripomínal pohyb „šijacieho stroja“, teda vykonával pohyb po sínusovke s amplitúdou 10 mm a periódou 1 s. Video som priložil ako prílohu `video_sinus.mp4`



Obrázok 14 - Ukážka kódu vo WorkVisual 6.0

4.4. Vytvorenie a nahratie programu na korekcie dráhy podľa dát zo snímača (snímač umiestnený paralelne ku smeru pohybu)

Keďže je snímač priamo napojený na cuppler robota, môžem ho vidieť na pendante medzi analógovými vstupmi pod číslom 7. Ručne si môžem odskúšať, že snímač mi vracia hodnoty 0-0,5, podľa toho v akej vzdialenosti je predmet. Hodnota nula predstavuje predmet v minimálnej snímačej vzdialenosti, hodnota 0,5 v maximálnej. Vstupné hodnoty sú s presnosťou na 3 desatinné miesta, čo nám v podstate definuje presnosť korekcií. Pokiaľ je celý snímaný rozsah 280 mm a vstupná hodnota do korekcií je s presnosťou na 3 desatinné miesta, predstavuje to 500 rôznych hodnôt, teda maximálna presnosť korekcie je (alebo teda skôr rozlíšiteľnosť zmeny odchýlky) bude 0,56 mm. Ďalším mojím zistením pozorovaním vlastností snímača je, že pokiaľ dôjde k extrémne rýchlemu odstráneniu predmetu (napríklad robot prejde sponad pásu, ktorý snímá nad zem), ostane snímač ukazovať hodnotu ktorú mal aj pred touto zmenou. Toto osobne považujem za výhodu, pretože pokiaľ by počas korigovania dráhy došlo k vybočeniu snímača mimo snímanej plochy, nedôjde k jeho nekontrolovateľnému pohybu smerom nadol ale ostane v pôvodnej hladine a zabráni sa tým pádom deštrukcií.



Obrázok 15 - umiestnenie snímača od TCP

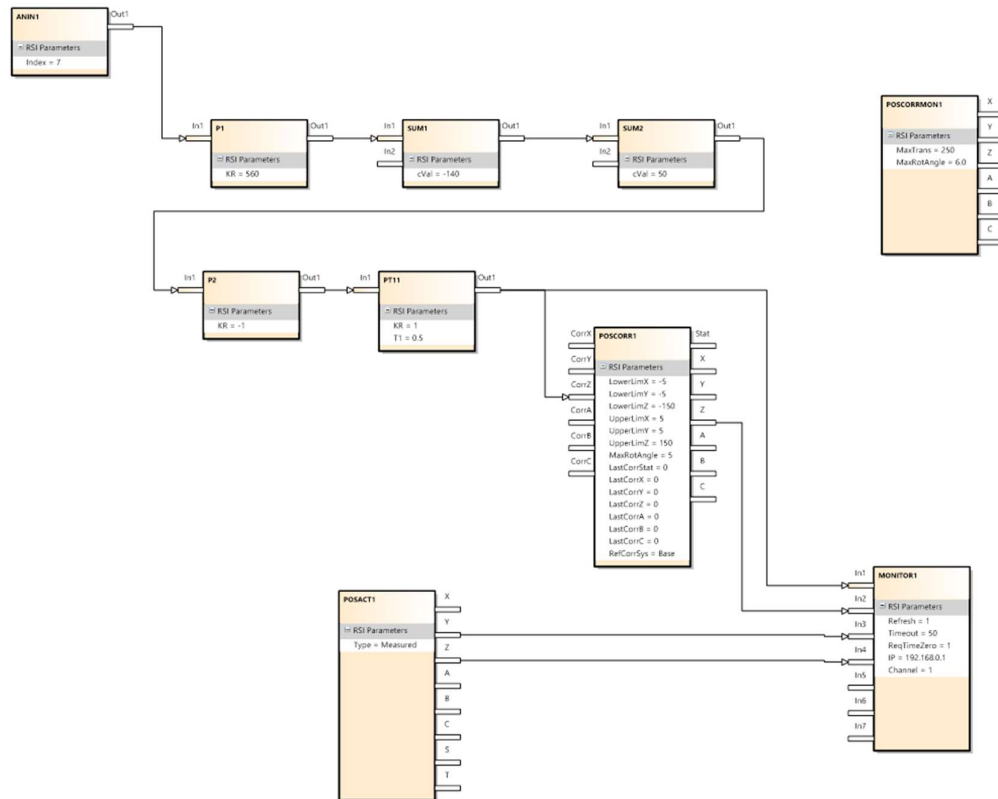
4.4.1. Program v RSI

Pre použitie analógového vstupu na korekcie v programe RSI vytvoríme blok s názvom ANIN (Analog Input), ktorému nastavím ako parameter číslo analógového vstupu, na ktorý má robot snímač pripojený. V mojom prípade teda vstup číslo 7. Následne je potrebné hodnoty zo snímača previesť na milimetre, s ktorými bude následne pracovať blok korekcií. Pokiaľ viem že hodnota 0,5 predstavuje maximálnu možnú hodnotu korekcie, teda 280 získam prevodový koeficient delením maximálneho rozsahu korekcie maximálnym rozsahom vstupu. Teda:

$$k = \frac{\text{Maximálny rozsah korekcie [mm]}}{\text{Maximálny rozsah vstupných hodnôt}} = \frac{280}{0.5} = 560$$

Pokiaľ vynásobíme vstupný signál hodnotou 560, získame hodnotu vzdialenosti predmetu od minimálnej snímacej vzdialenosti hlavice. Na násobenie použijem blok GAIN (alebo aj blok P), ktorému nastavím parameter 560. Aby sme ale mohli robiť korekcie aj v kladnom a zápornom smere, je nutné posunúť túto hodnotu na referenčnú hladinu snímača, teda o 140 mm. Toto spravím pomocou bloku SUM ktorému nastavím parameter -140. Tento blok teda odčíta od hodnoty korekcie hodnotu 140mm. V praxi to znamená, že pokiaľ bude predmet v referenčnej vzdialenosti 300 mm od snímacej

hlavice, vstup pre korekciu bude rovný nule. Pri vychýlení od tejto hodnoty bude hodnota záporná alebo kladná, podľa toho ktorým smerom sa predmet pohne. Tieto bloky slúžili priamo na spracovanie signálu. Pre užívateľskú prívetivosť som do schémy doplnil ešte jeden blok, ktorý predstavuje vzdialenosť TCP od referenčnej hladiny. Nie vždy totiž budeme mať hrot nástroja v tejto vzdialenosti. Dokonca aj v prípade hrotu, ktorý je na manipulátor možné namontovať, je nutné posunutie o 30 mm, a ešte aj z hľadiska bezpečnosti o ďalších 20, aby sa hrot pohyboval nad testovacou dráhou. Nastavím teda parameter tohto druhého SUM bloku na 50. Na záver je nutné použiť ešte jeden GAIN (P) blok, pretože vstupom do bloku korekcie je záporná väzba. Takýto signál už máme spracovaný, avšak korekcie by nám aj tak nefungovali. Nedostatok si môžeme všimnúť na RSI Monitore a spočíva v tom, že signál je veľmi šumivý a sám o sebe, hoci by išlo o prejazd ponad rovnú plochu, by nám robil veľmi rýchle malé korekcie. Toto je samozrejme softwarovo obmedzené v manipulátore, ktorý nám hneď hlási prekročenie maximálnych možných momentov a rýchlostí prevodoviek. Toto je možné odstrániť pridaním filtra, jednoduchého PT bloku, ktorému nastavíme parameter 0.5. Tento parameter je možné upravovať podľa zvlnenie vstupného signálu a rýchlosti pohybu, takže je potrebné sa s tým trochu pohrať a odladiť. Je potrebné, aby bol blok filtra až po pričítavacích blokoch, pretože by pri nábehu signálu spôsobil oneskorenie signálu a tým pádom by hneď na začiatku programu spôsobil korekciu o referenčnú hodnotu, čo vyvrcholí prekročením maximálnych možných momentov a rýchlostí prevodoviek a teda chybou programu. Schému zase doplním o bloky POSCORRMON, MONITOR a POSACT, ktoré som už v predchádzajúcej podkapitole opisoval a ich použitie je totožné. Na monitor si vytiahnem signáli zo snímača, signál vstupujúci do bloku korekcie, signál o aplikovanej korekcií a aktuálne polohy manipulátoru v osách Y a Z. Znova 3 uložené súbory uloží do priečinku C:\KRC\Roboter\Config\User\Common\SenzorInterface.



Obrázok 16 - ukážka RSI na korekcie na základe dát zo snímača

4.4.2. Robotický program

Vytvorím si ďalší základný program na lineárny pohyb medzi dvomi bodmi vo WorkVisual. Príkazom `ret=RSI_CREATE("navrh_pracoviste.rsi")` zadefinujem súbor s dátami o priebehu spracovania korekcií a príkazom `ret=RSI_ON()` ich aktivujem. Na záver vypnem príkazom `ret=RSI_OFF()`.

```

1  &ACCESS RVP
2  &REL 22
3  DEF test_pracoviste ( )
4  DECL INT ret
5  IN[ ]
16
17  HALT
18  ;koment
19  PTP P1 VEL=30 % PDAT1 TOOL[1] BASE[0]
26
27  ret=RSI_CREATE("navrh_pracoviste.rsi")
28  ret=RSI_ON()
29  ;WAIT SEC 20
30  LIN P3 VEL=0.1 M/S CPDAT1 TOOL[1] BASE[0]
37
38  ret=RSI_OFF()
39  ;test
40
41  PTP P2 VEL=30 % PDAT3 TOOL[1] BASE[0]
48  LIN P4 VEL=0.2 M/S CPDAT2 TOOL[1] BASE[0]
55  PTP P1 VEL=30 % PDAT2 TOOL[1] BASE[0]
62
63  END

```

Obrázok 17 - kód vo WorkVisual 6.0 pre korekcie zo snímačom

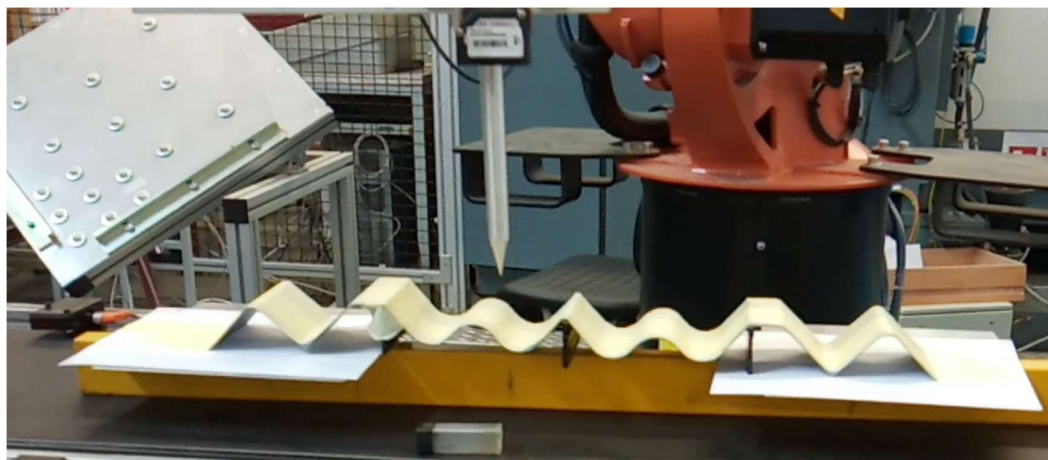
Ako skúšobnú dráhu som použil ľubovoľne navrhnutú dráhu ktorú som si vytlačil na 3D tlačiarňu. Dráha má tvar podľa obrázku, dĺžku 720 mm a obsahuje zaoblenia aj hrany.

Pri mojich pokusoch na dielni som zistil, že napriek použitiu filtru je signál veľmi citlivý a reaguje na malé zvlnenie tlačeneho predmetu. Pri vyšších hodnotách zase robot nestíhal reagovať vo vyšších rýchlostiach. Rovnako dochádzalo ku zlému snímaniu v miestach spojov, preto sme sa rozhodli s mojím školiteľom opatriť povrch papierovou páskou, ktorá všetky spomínané nedostatky odstráni. Problém by sa dal pravdepodobne vyriešiť aj použitím externého filtra zapojeného ešte pred vstup do zbernice manipulátoru.

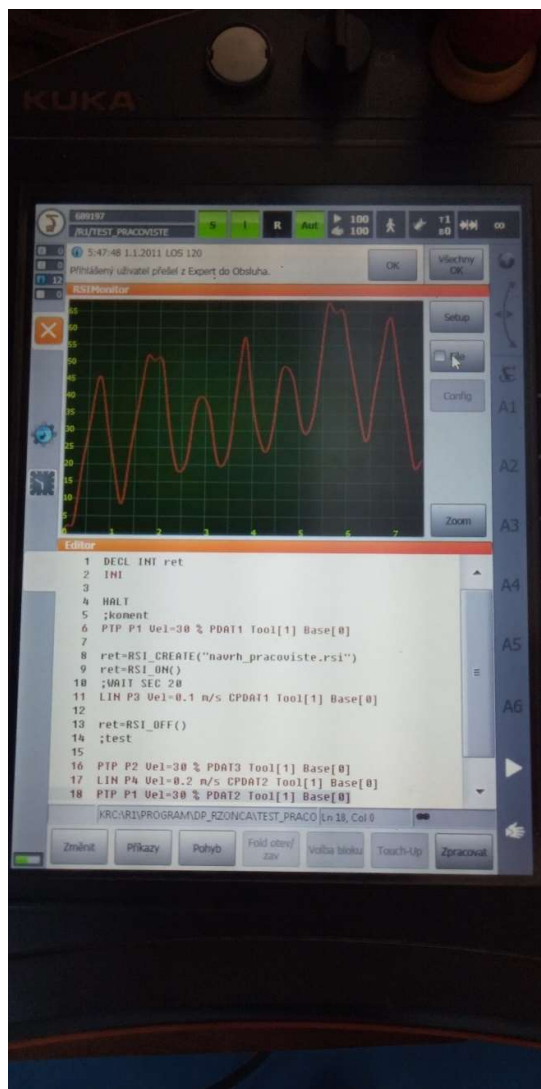


Obrázok 18 - Použitá testovacia dráha po úprave

Po tejto úprave bol už pohyb robota ponad dráhu ideálny a mohli sme robiť korekcie dráhy aj pri rýchlosti pohybu 0.1m/s. Ešte pri nastavovaní prvého bodu je nutné, aby bol tento bod aspoň približne v referenčnej hladine snímača a robot dostával údaje o korekcií približne 0. Nie je úplne nutné aby to bolo presné, ale pokiaľ je tento bod veľmi vzdialený od referenčnej hladiny, robot sa snaží robiť veľmi rýchlu korekciu o veľkú dráhu a dôjde k jeho zastaveniu z hľadiska bezpečnosti. Je to však ale dobrá skúška funkčnosti Safety blokov. Výstupné hodnoty som si v formáte .dat vyexportoval a spracoval v grafe signál zo snímača keyence, vstupný signál do korekcie aj signál s údajom o polohe. Video z priebehu korekcie som rovnako priložil ako prílohu video_senzor_1.mp4.



Obrázok 19 - Ukážka z odladovania korekcií podľa dát zo snímača

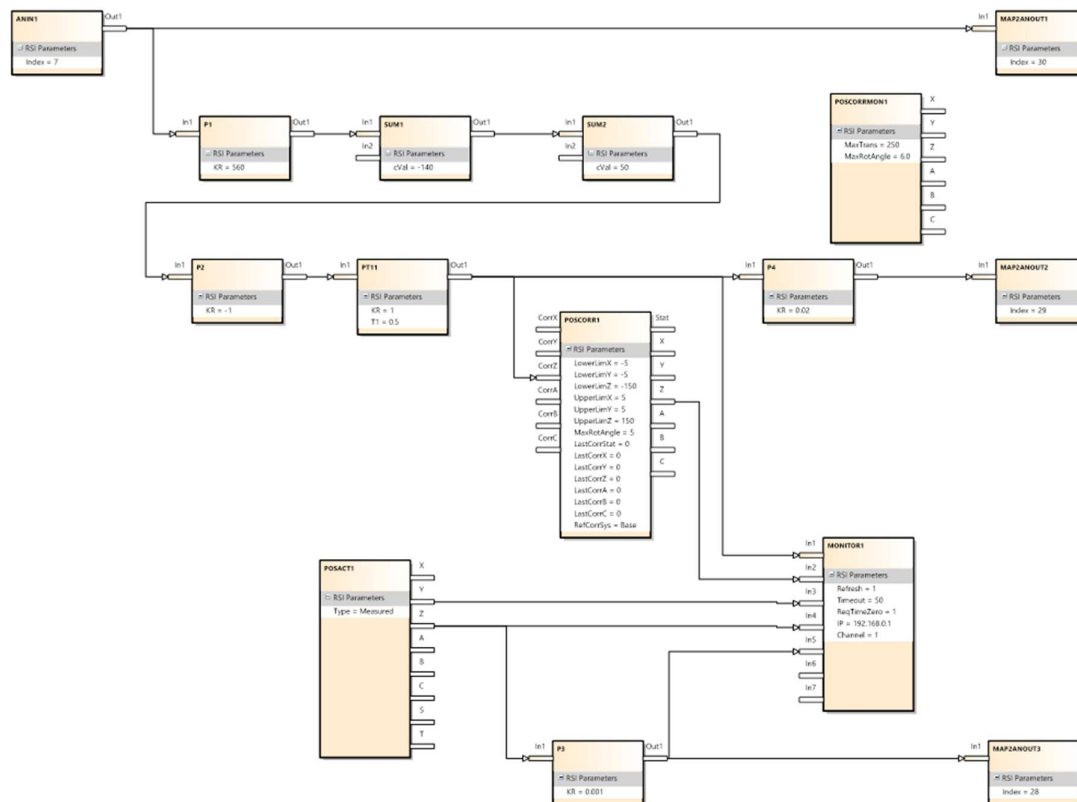


Obrázok 20 - Ukážka zobrazovaných dát RSI Monitorom na teachpendante

4.5. Doplnenie programu o HMI

Program je ešte nutné doplniť o zobrazovanie základných údajov o korekcií na PLC, ktoré sa nachádza na pracovisku. Toto PLC je pripojené do rovnakej siete ako robot a sú k nemu pripojené výstupné analógové signály robota. Pre komunikáciu medzi KR16 a PLC je potrebné najskôr zadať výstupné signály z RSI. Toto je možné pomocou bloku MAP2ANOUT, na ktorý sa privedie požadovaný výstupný signál a ako parameter sa nastaví číslo výstupného analógového signálu. Ako prvý som sa rozhodol vyviesť signál zo snímača, ktorý je v rozsahu 0-0.5. Maximálny výstupný rozsah na analógovom signáli je 0-10V. Ako ďalší by som chcel na HMI zobrazovať signál vstupujúci do bloku korekcií. Keďže tento môže nadobúdať hodnoty 0-420, predelím ešte výstupný signál číslom 500. Na toto použijem znovu blok GAIN (P), ktorému nastavím parameter 0.02. Tým výstupný parameter zmenším, aby jeho hodnota bola v rozsahu 0-10, a teda v reálne dosiahnuteľných hraniciach analógového výstupu. Tento signál spojím s ďalším blokom MAP2ANOUT, ktorému nastavím parameter 29, čo reprezentuje číslo analógového výstupu. Tretí signál, ktorý by som chcel zobrazovať, je údaj o polohe robota v ose Z. Tento je tak isto z rozsahu

0-1000, preto výstupnú hodnotu prenášobím pomocou P bloku koeficientom 0.02. S takto upravenými signálmi môžem pracovať v TwinCate.



Obrázok 21 - upravený RSI kód pre výstup na HMI

Na PLC použijem už vytvorený projekt do ktorého doplním vizualizáciu, nadefinujem premenné a doplním blok na ich spätný prepočet. Najskôr je potrebné vytvoriť globálne premenné, teda také ku ktorým má prístup celé premenné a sú vytvorené od začiatku programu až po jeho koniec (sú teda trvalo alokované v pamäti). Pre každý signál, ktorý chcem prenášať na monitor si vytvorím jednu dvojicu (1 INT a 1 REAL), prvú premennú na vstupné údaje a druhú na prácu s HMI. Medzi týmito dvoma premennými bude ešte dochádzať k prepočtu vo funkčnom bloku. Analógové vstupy sú už nalinkované, je potrebné akurát nalinkovať ktorá premenná bude na ktorom vstupe.

Variable	Flags	Online
Name:	Var 119	
Type:	INT	
Group:	IO Inputs	Size: 2.0
Address:	97 (0x61)	User ID: 0
Linked to...	GVL.Val_realna_hodnotaZ . PlcTask Inputs . PLC Instance . PLC	
Comment:		
ADS Info:	Port: 11, IGrp: 0x3040020, IOFs: 0x80000061, Len: 2	
Full Name:	TIID*Device 3 (EtherCAT)*Term 1 (EK1100)*Term 12 (EL6692)*IO Inputs*Vz	

Obrázok 22 - linkovanie premennej na analógový vstup

Následne môžem začať s premennými pracovať vo funkčnom bloku, kde si pre každú vytvorím ekvivalent v rámci bloku (na ktorý si prilinkujem vstup a výstup vo funkcii main príkazom: fbKeyence(nSnimacIn:= GVL.Val_zo_snimaca , rSnimacHMI=> GVL.ValHmi_zo_snimaca))

```

9 num:=KR1.nProgram;
10
11 fbKeyence(nSnimacIn:= GVL.Val_zo_snimaca , rSnimacHMI=> GVL.ValHmi_zo_snimaca);
12 fbKeyence(nKorekciaIn:= GVL.Val_do_korekcie , rKorekciaHMI=> GVL.ValHmi_do_korekcie);
13 fbKeyence(nPolohaIn:= GVL.Val_realna_hodnotaZ , rPolohaHMI=> GVL.ValHmi_realna_hodnotaZ);
14

```

Obrázok 23 - Doplnenie do main kódu

S internými premennými v bloku môžem začať pracovať, každú je potrebné správne prepočítať. Keďže vstup do PLC je typu INT, vstupné hodnoty sú v rozsahu 0 až 32 767, čo predstavuje rozsah 0-10 V na analógovom vstupe. Preto je teda potrebné vstupnú hodnotu podeliť celým rozsahom, čím získame percentuálny podiel vstupnej hodnoty z jej maximálnej možnej hodnoty. Následne už iba prenášobíme mierkou.

$$rSnimacHMI = \frac{INT_TO_REAL(nSnimacIn)}{32767.0} * 5.0$$

Samozrejme treba mať na pamäti, že pokiaľ pracujeme s hodnotami formátu REAL a INT, je potrebné ich medzi sebou previesť príkazom INT_TO_REAL.


```

1 FUNCTION_BLOCK FB_KEYENCE
2 VAR_INPUT
3     nSnimacIn      : INT;
4     nKorekciaIn    : INT;
5     nPolohaIn      : INT;
6
7 END_VAR
8 VAR_OUTPUT
9     rSnimacHMI      : REAL;
10    rKorekciaHMI     : REAL;
11    rPolohaHMI       : REAL;
12 END_VAR
13 VAR
14
15 END_VAR
1
2 rSnimacHMI:=(INT_TO_REAL(nSnimacIn)/16383.0)*5.0;
3 rKorekciaHMI:=(INT_TO_REAL(nKorekciaIn)/32767.0)*500.0;
4 rPolohaHMI:=(INT_TO_REAL(nPolohaIn)/32767.0)*1000.0;

```

Obrázok 24 - vytvorený funkčný blok na prepočítavanie hodnôt zo vstupov

Blok prepočtu je týmto hotový a môžeme hodnotu preniesť naspäť na GLV, odkiaľ sa už bude zobrazovať rovno na bar grafe. Tu sa vstupná hodnota nadefinuje priamo ako vstupný parameter bar grafu. [10]

Filter • Sort by • Sort order • Advanced	
Property	Value
Element name	GenElemInst_2
Type of element	Bar display
Value	GVL.ValHmi_zo_snimaca
Position	
X	277
Y	164
Width	64
Height	539
Background	

Obrázok 25 - linkovanie GVL premennej na HMI

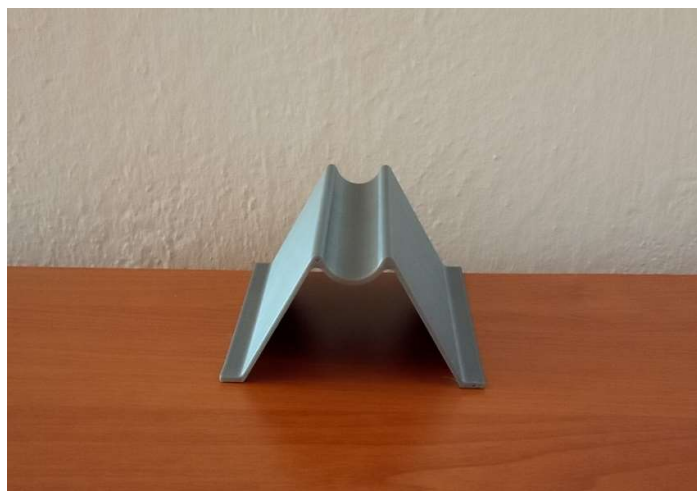


Obrázok 26 - Vytvorené HMI na zobrazovanie dát o korekcií

Veľkou výhodou zosieťovaného PLC je možnosť pracovať s ním online na vlastnom PC, takže som si mohol na začiatku stiahnuť počítačové údaje, nastaviť a naprogramovať všetko potrebné na vlastnom PC a následne to znova nahráť do PLC. Tento postup je oveľa prívetivejší a jednoduchší ako pracovať priamo na PLC v dielni.

4.6. Vytvorenie a nahratie programu na korekcie dráhy podľa dát zo snímača (snímač umiestnený pred TCP)

V poslednom programe som sa rozhodol vytvoriť program na korekcie, kedy robot prejde dráhu najskôr snímačom a až potom bude vykonávať korekcie, teda s dopravným oneskorením. Najskôr som sa rozhodol takýto program spraviť pre krátku dráhu, kedy dôjde k nasnímaniu celej dráhy a následne k jej skopírovaniu. Vzdialenosť medzi bodom snímání a špičkou hrotu je 80 mm, preto spomínaný predmet musí byť do maximálne takejto šírky. Pokiaľ by bol predmet širší, po začatí vykonávanie korekcie by si robot do korekcií započítaval aj svoje vlastné pohyby. To by spôsobilo nefunkčnosť korekcií.



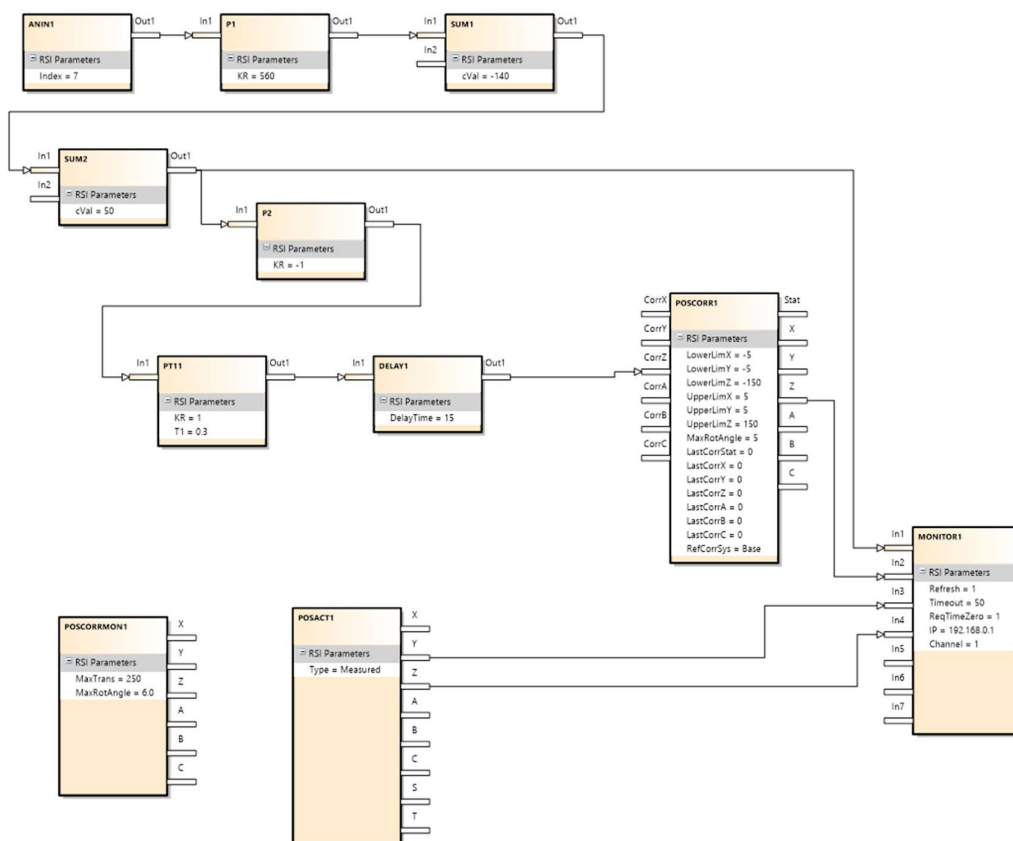
Obrázok 27 - Použitá dráha pre snímání s dopravným oneskorením

4.6.1. Program v RSI

Na vytvorenie programu som použil rovnaký základ, aký som mal už odskúšaný pri paralelnom umiestnení snímača. Blok vstupnej hodnoty ANIN, blok GAIN(P) na prevod signálu na milimetre aj dva SUM bloky môžu ostať nezmenené. Rovnako môžem použiť aj filter a druhý P blok na vytvorenie zápornej väzby. Keďže ide snímač dopredu, musíme signál vstupujúci do korekcií oneskoriť o rozdiel medzi bodom dopadu lúča a špičkou hrotu. Žiaľ, RSI neobsahuje žiadny blok na vzdialostné oneskorenie, iba na časové. Z rýchlosti pohybu 0,005 m/s a vzdialenosti 80 mm si môžeme vypočítať čas oneskorenia:

$$t = \frac{s}{v} = \frac{0,08 \text{ m}}{0,005 \text{ m} * \text{s}^{-1}} = 16 \text{ s}$$

Vložím teda do schémy ešte blok DELAY s parametrom 16, ktorý mi značí oneskorenie signálu o 16s. Túto schému znova uloží a tri súbory nahrám do priečinka C:\KRC\Roboter\Config\User\Common\SenzorInterface.

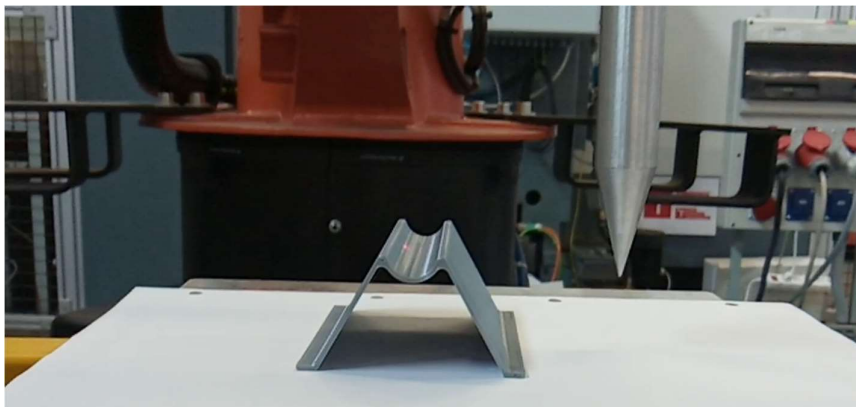


Obrázok 28 -RSI kód pre snímanie korekcií s dopravným oneskorením

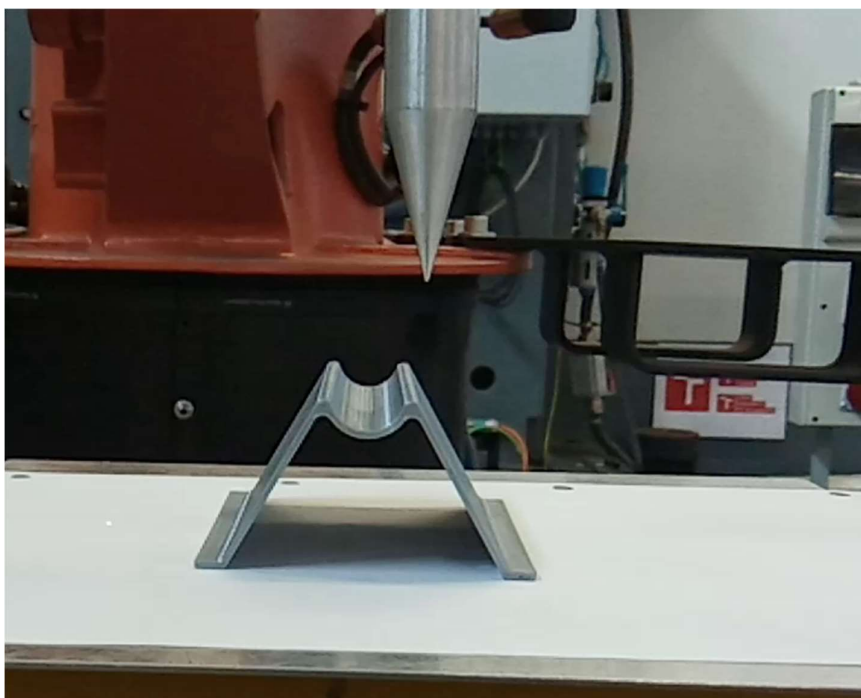
4.6.2. Robotický program

Tento program bude znova zložený iba z lineárneho pohybu medzi dvomi bodmi, ktoré si ručne nájdem a prepíšem pomocou funkcie TeachUP. Druhý bod je umiestnený za prekážkou čo najbližšie, aby nedošlo k aplikácii korekcií ktoré by boli ovplyvnené pohybom robota po korigovanej dráhe (tak ako som už písal v podkapitole vyššie).

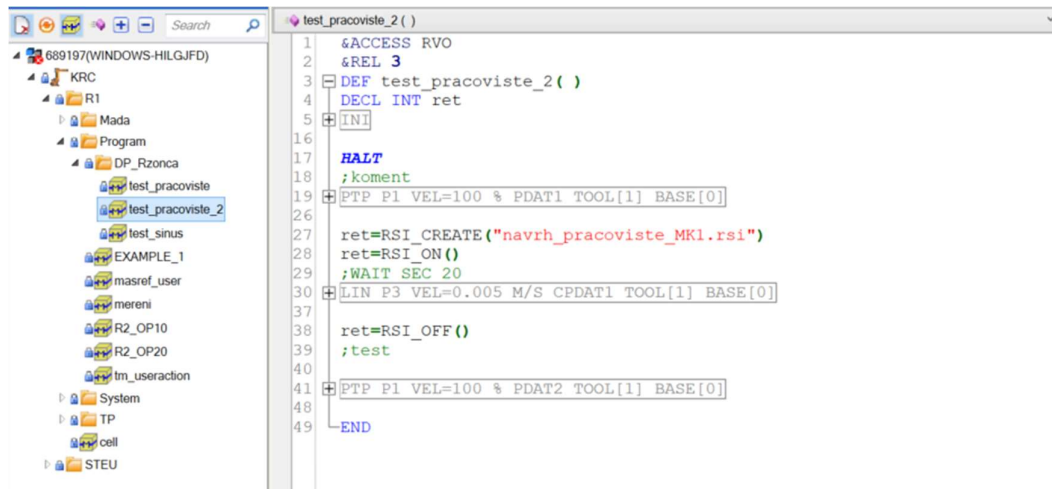
Príkazom `ret=RSI_CREATE("navrh_pracoviste_MK3.rsi")` načítam dáta o korekciách, príkazom `ret=RSI_ON()` zapnem RSI ešte pred samotným korigovaným pohybom a po vykonaní požadovanej dráhy príkazom `ret=RSI_OFF()` korekcie vypnem. Video z priebehu korekcie som rovnako priložil ako prílohu `video_senzor_2.mp4`



Obrázok 29 - ukážka nasnímania hodnoty korekcií



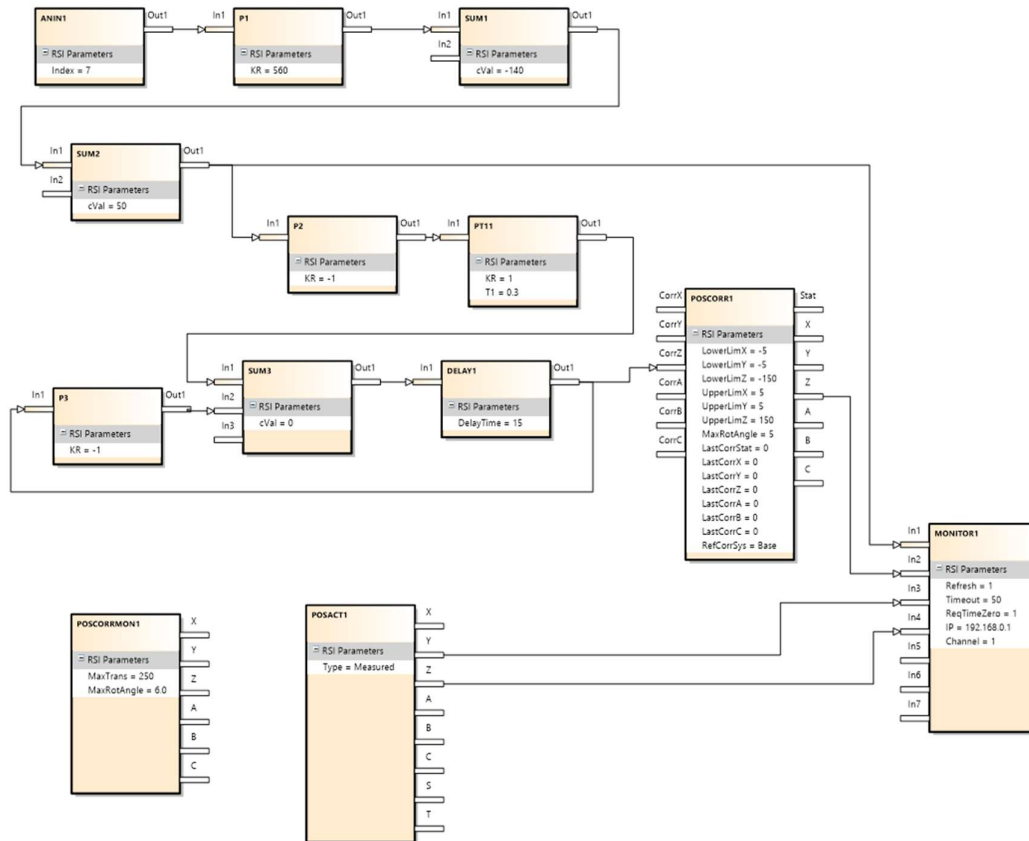
Obrázok 30 - ukážka aplikácie korekcií s dopravným oneskorením



Obrázok 31 - vytvorený kód vo WorkVisual 6.0 pre korekcie s oneskorením

4.6.3. Úprava programu pre dlhšie dráhy

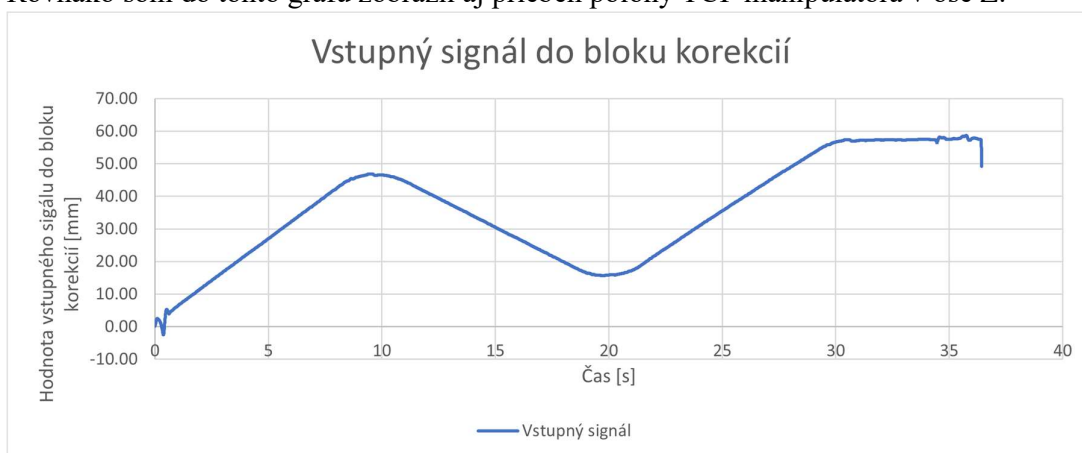
Pokiaľ by sme chceli aplikovať korekcie, pri ktorých pôjde senzor v časovom predstihu pred robotom, je nutné aby sme odstránili chybu, ktorú si robot sám spôsobí pri pohybe po korigovanej dráhe. Tento nedostatok spočíva v tom, že ak robot začne stúpať smerom hore, vzdialenosť medzi senzorom a podložkou sa mu nechcane mení. Je teda potrebné odčítať od hodnoty zo senzora hodnotu korekcie, ktorá bola nameraná pred 16 sekundami, a teda práve tú zmenu o ktorú je TCP posunuté. Z tohto dôvodu som použil ďalší SUM blok, ktorý som vložil pred DELAY blok a tým vytvoril túto spätnú väzbu. Schému som doplnil ešte o jeden GAIN(P) blok, ktorý mi vracia zápornú hodnotu aktuálnej korekcie, pretože RSI neobsahuje blok na odčítanie, iba na sčítanie. Nie je ani možné napojiť do schémy výstupný signál z POSCORR bloku, ktorý by nám mal poskytnúť informácie o aktuálne vykonanej korekcií, pretože v takomto prípade zahlásí robot po spustení programu chybu v RSI a vôbec neaktivuje RSI Container v ktorom sú údaje o korekcií uložené. Začne sa teda pohybovať po nekorigovanej dráhe. Schému pre takéto korekcie sa mi podarilo vytvoriť, žiaľ nepodarilo sa mi ju odskúšať na pracovisku, pretože ako študent zo Slovenska som nemal umožnené ubytovanie na internátoch a teda som musel prichádzať do dielne vždy len na otočku a tým pádom aj na relatívne krátky čas. Aj tak som sa ju ale rozhodol priložiť, pretože mi prišla ako jedna z najvyužiteľnejších v praxi, keďže je ňou možné snímať priamo dráhu po ktorej hrot pôjde. Verím, že ešte v budúcnosti budem mať možnosť si ju odskúšať.



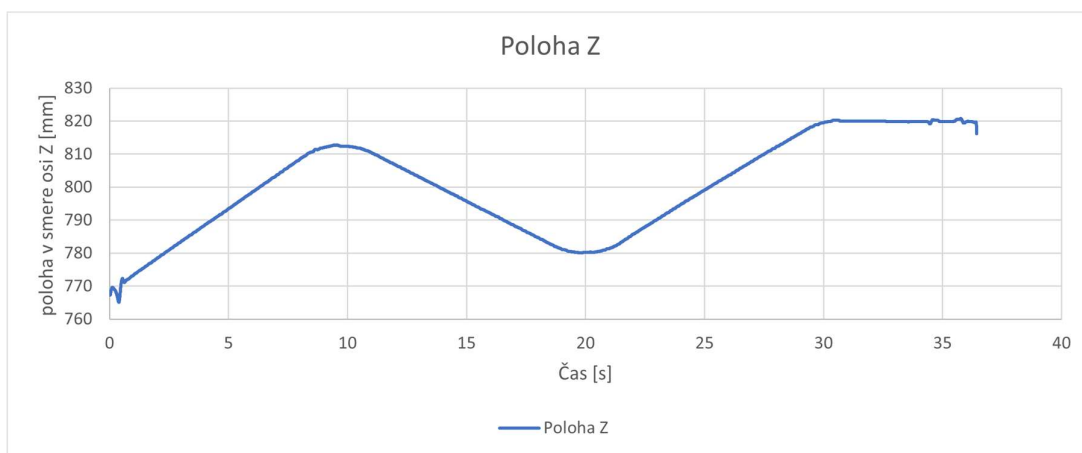
Obrázok 32 - Upravený RSI program pre korekcie s oneskorením a spätnou väzbou

5. Vyhodnotenie údajov z RSI Monitoru

Po zapojení signálov v RSI Visual do bloku Monitor si môžeme tieto údaje na touchpade nielen zobrazit', ale ručne aj stiahnuť do .dat súboru. Súbor sa uloží do zložky C:\KRC\Roboter\Log. Keďže som si aj tento priečinok zasiet'oval, mohlo som si ich jednoducho stiahnuť do PC a spracovať. Získané dáta môžeme využiť najmä pokiaľ niekde na dráhe dochádza k chybe, ako sa to stávalo aj v mojom prípade pri prechode cez hranu spoja dráhy. Rozhodol som sa preto tieto dáta spracovať do grafu, kde môžeme vidieť priebeh signálu vstupujúceho do bloku korekcií. Na prvých dvoch grafoch som zobrazil dáta o veľkosti korekčného signálu a polohe TCP v ose Z pred aplikáciou papierovej pásky, na druhom po jej aplikácii. Prvé grafy sú skrátené na konkrétny úsek, pretože na poslednom viditeľnom bode došlo k prekročeniu maximálnej možnej rýchlosti a momentu prevodovky manipulátoru, na druhej dvojici je zobrazený celý priebeh tohto signálu počas celej dráhy. Rovnako som do tohto grafu zobrazil aj priebeh polohy TCP manipulátora v ose Z.



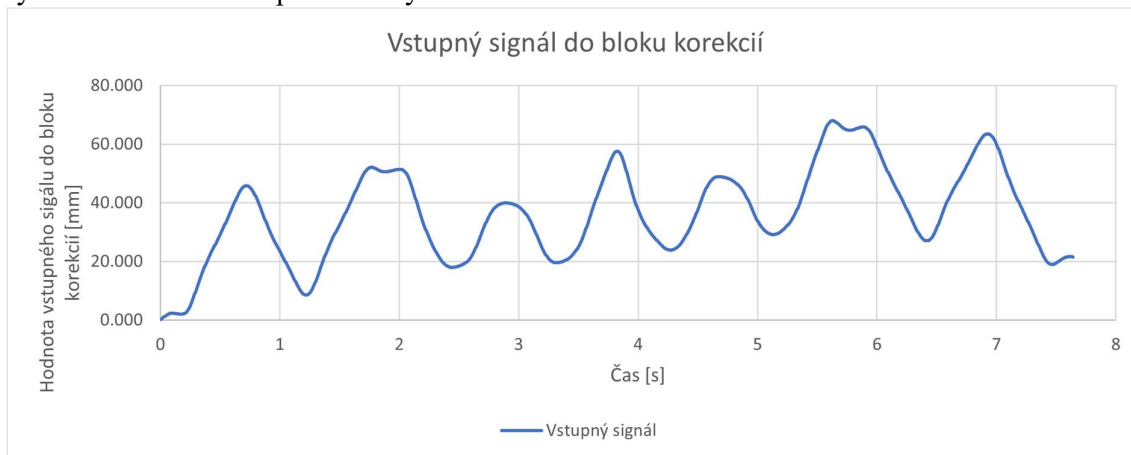
Graf 1 – časová závislosť vstupného signálu do bloku korekcií



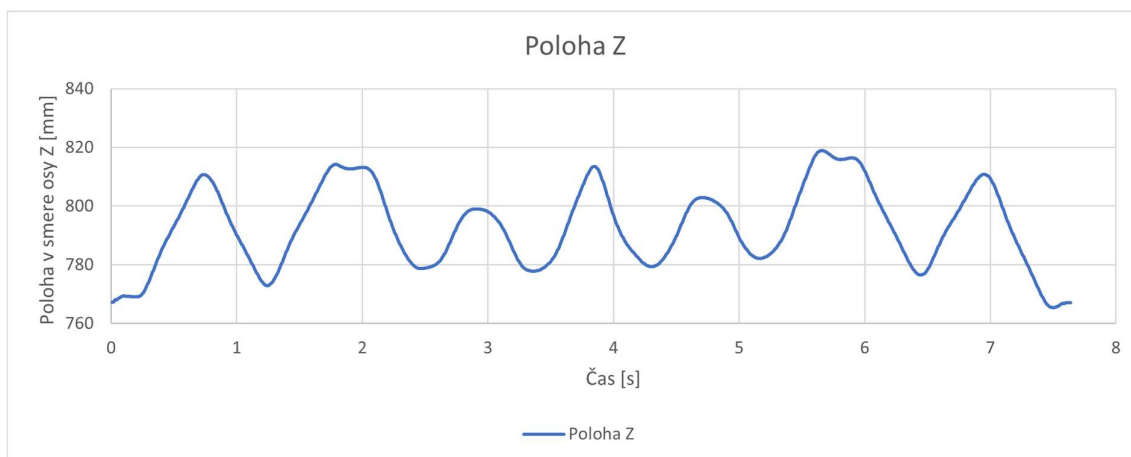
Graf 2 - časová závislosť polohy TCP v ose Z

Tieto dva grafy boli zhotovené počas prejazdu ponad dráhu, ktorá ešte nebola opatrená papierovou páskou. Môžeme vidieť chybné odrazy na začiatku dráhy, kedy sa to prejavilo zakmitaním TCP v ose Z a rovnako došlo k tejto chybe aj na konci, kde senzor našiel na

hranu kde boli dráhy spojené a tu už bola chyba taká kritická, že došlo k výraznému poklesu v smere osy Z a robot sa zastavil, pretože došlo k prekročeniu maximálnych možných rýchlostí a momentov prevodovky.



Graf 3 - časová závislosť vstupného signálu do bloku korekcií



Graf 4 - časová závislosť polohy TCP v ose Z

Naopak tieto dva záznamy boli zhotovené pri prejazde ponad dráhu ktorá bola opatrená papierovou páskou a kde nedochádzalo ku zlým odrazom, preto je dráha výrazne hladšia a robot ju dokázal prejsť celú. V tomto prípade sme teda mohli pristúpiť aj k zvýšeniu prejazdovej rýchlosti a dokonca spustiť manipulátor v automatickom režime. Môžeme však ale vidieť, že dáta zo snímača vzdialenosti, ale aj dáta o polohe, nie sú úplne presné a občas dochádzalo k jemným odchýlkam.

Záver a zhodnotenie výsledkov práce

Z prieskumu trhu v prvej časti práci je zjavné, že každý väčší výrobca manipulátorov k nim ponúka opcii, ktorá umožňuje zákazníkovi robiť korekcie dráhy online na základe rôznych vstupných parametrov. Týmto môžu byť buď priamo vstupné parametre rôznych senzorov, alebo aj iné parametre dopočítané napríklad pomocou PLC z dát z iného manipulátoru alebo zariadenia, ale aj iné získané z virtuálneho dvojčaťa pracoviska. Ponúkané varianty sa líšia svojimi funkciami podľa výrobcu, ale aj robustnosťou opcie a spôsobom práce s ňou. Opcia RSI od výrobcu KUKA je užívateľsky najprívetivejšia, práca s ňou je vďaka rôznym doplnkovým programom relatívne jednoduchá. Opcia Alter od výrobcu Stäubli je jednoduchá a programátorsky nenáročná, bez rôznych doplnkových funkcií. Pokiaľ zákazník potrebuje iba korigovať dráhu podľa vstupných dát, je asi najlepším riešením. Naopak opcia EGM od výrobcu ABB patrí medzi najrobustnejšie a okrem funkcií korekcií obsahuje aj ďalšie iné, čo je vhodné napríklad pre zákazníkov, ktorí plánujú nasádzať svoje roboty na rôznych pracoviskách a programy a funkcie robotov budú často menené. Zvolená varianta môjho riešenia bola volená najmä s ohľadom na dostupné možnosti v dielni, kde je dostupný 6-osý manipulátor od výrobcu KUKA typu KR16 s opciou RSI, ktorá už bola zakúpená a nainštalovaná pred začiatkom mojej práce, rovnako ako snímač vzdialenosti Keyence IL-300 a zosilňovač Keyence IL-1000. Tieto parametre značne zjednodušili výber a teda som praktickú časť mojej práce vypracoval práve na tomto manipulátore. K programovaniu som využil rozhranie RSI Visual ktoré si je možné nainštalovať do PC a pracovať s ním aj mimo dielne. Môžem zhodnotiť že práca v ňom je intuitívna a nemal by mať s ňou problém ani integrátor – začiatočník. Bloky sú prehľadne usporiadané a ich funkcie jasne popísané. Nahratie do robota je tiež jednoduché a jasné. Program je však vždy potrebné poriadne odladiť pri nižších rýchlostiach, pretože okrem chyby v programe môže tiež dôjsť ku zlému signálu alebo odrazu a zlému spracovaniu dát zo senzoru, čo môže spôsobiť aj poškodenie manipulátoru. Rovnako nemusí vždy bloková schéma fungovať úplne podľa predstavy, preto je potrebné vždy všetko poriadne odladiť, hlavne pri nových aplikáciách. Po úspešnom odladení a prípadných malých úpravách bolo možné robiť korekcie aj vo vyšších rýchlostiach. Podarilo sa mi úspešne robiť korekcie pokiaľ si robot snímал vzdialenosť paralelne vedľa špičky hrotu aj na dlhších dráhach, cca 700 mm, ale aj pokiaľ si ju nasnímal vopred a korekcie robil s oneskorením. Toto sa mi však podarilo odladiť len na kratšej dráhe, približne 80 mm, na dlhších dráhach som túto funkciu nestihol odskúšať. Rovnako úspešne sa mi podarilo sfunkčnúť komunikáciu a prenos údajov z kontroléru manipulátoru do PLC a ich zobrazovanie na obrazovke. Online korekcie dráhy môžu patriť medzi obľúbené doplnkové vlastnosti manipulátorov najmä s rozvojom autonómnych systémov a priemyslu 4.0, pretože umožňujú manipulátoru robiť real-time zásahy do jeho dráhy, čo je v základnom programovaní manipulátoru zakázané a manipulátor musí vždy pred zmenou dráhy dokončiť. Získané skúsenosti sa mi dúfam podarí využiť aj v ďalšom štúdiu na Vysokom Učení Technickom v Brne.

ZOZNAM POUŽITÝCH ZDROJOV

- [1] KUKA. *RobotSensorInterface 3.3 for KUKA System Software 8.3 and 8.4*. KUKA Roboter GmbH. 2014
- [2] Stäubli. *Stäubli* [online]. [cit. 2021-5-20]. Dostupné z: <https://www.staubli.com/en/robotics/product-range/industrial-robots/6-axis-robots/>
- [3] VAL 3 REFERENCE MANUAL: Version 8. Stäubli International AG. 2019
- [4] ABB. *ABB* [online]. [cit. 2021-5-20]. Dostupné z: <https://new.abb.com/products/robotics/cs/prumyslove-roboty/irb-2400>
- [5] *Application manual -Externally Guided Motion*. 2020.
- [6] *Keyence IL-300 datasheet*. KEYENCE INTERNATIONAL. 2015.
- [7] *Keyence IL-1000 datasheet*. KEYENCE INTERNATIONAL. 2015.
- [8] Beckhoff EL3062. *Beckhoff* [online]. [cit. 2021-5-20]. Dostupné z: <https://www.beckhoff.com/sk-sk/products/i-o/ethernetcat-terminals/el3xxx-analog-input/el3062.html>
- [9] *Keyence IL Series Instruction Manual*. KEYENCE CORPORATION. 2015.
- [10] TwinCAT 3 – eXtended Automation Technology (XEA) [online]. Automa, 2016 [cit. 2021-5-20]. Dostupné z: https://automa.cz/cz/casopis-clanky/twincat-3-extended-automation-technology-xat-2015_06_53801_6979
- [11] SPURGEON, Charles E. a Joann ZIMMERMAN. *Ethernet: The Definitive Guide: Designing and Managing Local Area Networks*. 2. O'Reilly Media, 2014. ISBN 9781449361846.

ZOZNAM POUŽITÝCH SYMBOLOV, JEDNOTIEK A SKRATIEK

Symbol	Veličina	Jednotka
t	Čas	[s]
s	Vzdialenosť	[m]
v	Rýchlosť	[m/s]

Symbol	Jednotka	Veličina
V	Volt	Napätie
mA	Miliampér	Prúd
Ω	Ohm	Odpor
mm	Milimeter	Vzdialenosť
μm	Mikrometer	Vzdialenosť
g	gram	Hmotnosť

ZOZNAM OBRÁZKOV

Obrázok 1 - ukážka .xml kódu RSI pre online korekcie	13
Obrázok 2 - Výmena dát v rámci RSI [1]	14
Obrázok 3 - Výmena dát cez A/D I/O porty [1]	15
Obrázok 4 - komunikácia cez Ethernet [1]	15
Obrázok 5 - 6-osý manipulátor Stäubli [2]	19
Obrázok 6 - 6-osý manipulátor ABB [4]	21
Obrázok 7 - a) KUKA KR16, b) PLC na pracovisku KUKA UVSSR	27
Obrázok 8 - a) snímač namontovaný IL-300 na KR16, b) detail na snímač [6]	28
Obrázok 9 a) zosilňovač IL-100 na KR16, b) detail na zosilňovač [7]	29
Obrázok 10 - karta Beckhoff EL3062 a schéma zapojenia [8]	29
Obrázok 11 - EL3062 zapojená na pracovisku	30
Obrázok 12 - Cuppler umiestnený pod KR16	30
Obrázok 13 - ukážka programu pre korekcie na základe sínusového signálu	32
Obrázok 14 - Ukážka kódu vo WorkVisual 6.0	33
Obrázok 15 - umiestnenie snímača od TCP	34
Obrázok 16 - ukážka RSI na korekcie na základe dát zo snímača	36
Obrázok 17 - kód vo WorkVisual 6.0 pre korekcie zo snímačom	36
Obrázok 18 - Použitá testovacia dráha po úprave	37
Obrázok 19 - Ukážka z odlaďovania korekcií podľa dát zo snímača	37
Obrázok 20 - Ukážka zobrazovaných dát RSI Monitorom na teachpendante	38
Obrázok 21 - upravený RSI kód pre výstup na HMI	39
Obrázok 22 - linkovanie premennej na analógový vstup	40
Obrázok 23 - Doplnenie do main kódu	40
Obrázok 24 - vytvorený funkčný blok na prepočítavanie hodnôt zo vstupov	41
Obrázok 25 - linkovanie GVL premennej na HMI	41
Obrázok 26 - Vytvorené HMI na zobrazovanie dát o korekciách	42
Obrázok 27 - Použitá dráha pre snímanie s dopravným oneskorením	42
Obrázok 28 - RSI kód pre snímanie korekcií s dopravným oneskorením	43
Obrázok 29 - ukážka nasnímania hodnoty korekcií	44
Obrázok 30 - ukážka aplikácie korekcií s dopravným oneskorením	44
Obrázok 31 - vytvorený kód vo WorkVisual 6.0 pre korekcie s oneskorením	45
Obrázok 32 - Upravený RSI program pre korekcie s oneskorením a spätnou väzbou	46

ZOZNAM TABULIEK

Tabuľka 1 - prehľad dostupných opcíí	13
--	----

ZOZNAM GRAFOV

Graf 1 – časová závislosť vstupného signálu do bloku korekcií	47
Graf 2 - časová závislosť polohy TCP v ose Z.....	47
Graf 3 - časová závislosť vstupného signálu do bloku korekcií.....	48
Graf 4 - časová závislosť polohy TCP v ose Z.....	48

ZOZNAM PRÍLOH

Príloha

video_sinus.mp4

video_senzor_1.mp4

video_senzor_2.mp4

sinusovka_DP.rsi

navrh_pracoviste.rsi

navrh_pracoviste_MK1.rsi

Measurement_01.dat

Measurement_05.dat